Video Game Sales with Ratings Prediction

OPIM 5894 Data Science with Python

Team Project Report

Mengmeng Zhai, Shan Xu, Yuhan Chen

Date: 9th December 2018

Section 1 Introduction

Video games have come a long way since the first games emerged. As the video game market is getting more and more flourishing, the video game industry is booming with continued revenue. Thus, the game developers and game players are increasingly concerned about video game sales. The sales will help the sales department to make effective business decisions and make the video games fans to choose popular games tailored to their taste.

The objective of our team project is to apply machine learning knowledge with python to explore key factors in predicting video game sales by using data obtained from Kaggle. In general, we will perform data prepossessing, exploratory data analysis, model building, model selection and providing business findings and business recommendations based on the previous analytics results.

Section 2 Motivation

Initially, the Video Games Sales data come from the Gregory Smith's web scrape of VGChartz. The motivation of this project is to combine the existing sales data with another web scrape from Metacritic, which is a website aggregates scores of different media products including video games, TV shows, and films. Their rating process typically involves the following key parameters:

- **User_Score**: The score given by subscribers from Metacritic.
- **User_Count**: The number of users who gave User_Score.
- **Critic_Score**: Aggregate score compiled by Metacritic staff.
- **Critic_Count**: The number of critics used in creating Critic_Score.

The above factors have been included in our dataset as well. Given the scores information from Metacritic, we think it's worthwhile for us to put the knowledge into the practice and do relative predictions on this video game sales predictions.

Section 3 Literature and Distinction

According to IBIS World Report, the Video Games industry has experienced strong growth which leads high business return over the past couple of years. The U.S. industry has grown by an annual rate of 7.2% to reach revenue of $44 billion in 2018, with the average company profit margin of 3.7 billion. It is not exaggerated to conclude that the video games industry has become one of the most promising industries in the united states, even to the world.

As illustrated above, the researching of the correlation between video games sales and video games features started by the web scrape from VGChartz Video Games Sales. Then, as there is some additional games rating information have been obtained from the Metacritic website, it allows people to extend the investigation by adding more factors which may have the potential correlation with the

video games sales.

The initial purpose of this Kaggle dataset is providing users with a real-world opportunity to learn data visualization using different python libraries, including Matplotlib and Seaborn. The box plots and scatter plots are mostly common being used to explore this dataset. Later, a couple of people made the prediction of global sales based on the game features and the rating information; and presented the model accuracy assessment accordingly.

According to the earlier work, our team want to employ some different approaches when making the prediction of global sales. To begin with, we decide to perform both regression and classification prediction for global sales using the same dataset because we are interested to see if this dataset is better fitted for global sales or the popularity based on the results. In addition, we have found the target class "Hit" is high imbalanced which only 15% of the observations are displayed as 1, this may easily cause the prediction bias. Therefore, we use the up-sample method to balance the binary target variable to increase the accuracy of prediction of "1". Another thing worth mentioning is that we have realized some of the features are quite important but include many categories, such as publisher and platform. Converting these variables to dummy variables will add hundreds of new variables into our dataset. Thus, we only keep the high-frequency dummy variables in order to decrease the model complexity, as well as keep the valuable information from those features.

Section 4 Data Overview

4. 1 Data Description

The dataset was obtained from Kaggle which name as video games sales with ratings. It includes 16719 observations with 16 variables, 6 of them are categorical. Each row represents the features and ratings for a particular video game. The information of this dataset can be divided as video games general information, the scores from Metacritic, and regional sales information (Japan/Europe/North America/Other countries). The below graph illustrates the detailed description and datatype for each variable.

Column name	Data type	Description
Name	String	Name of video game
Platform	String	Console on which the game is running
Year_Of_Release	Numeric	Year of the game released
Genre	String	Game's category
Publisher	String	Publisher
NA_Sales	Numeric	Game sales in North America (in millions of units)
EU_Sales	Numeric	Game sales in the European Union (in millions of units)
JP_Sales	Numeric	Game sales in Japan (in millions of units)

Other_Sales	Numeric	Game sales in the rest of the world, i.e. Africa, Asia excluding Japan, Australia, Europe excluding the E.U. and South America (in millions of units)
Global_Sales	Numeric	Total sales in the world (in millions of units)
Critic_Count	Numeric	The number of critics used in coming up with the Critic_score
Critic_Score	Numeric	Aggregate score compiled by Metacritic staff
User_Score	Numeric	Score by Metacritic's subscribers
User_Count	Numeric	Number of users who gave the user_score
Developer	String	Party responsible for creating the game
Rating	String	The ESRB ratings (E.g. Everyone, Teen, Adults Only..etc)

4.2 Descriptive Statistics

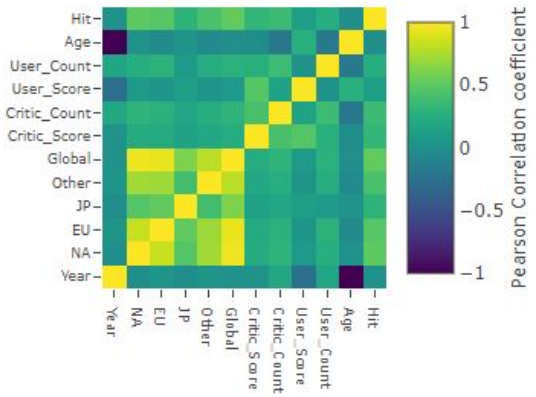
The basic statistic can help us take an overview of the dataset before preprocessing and exploration, we can see the average of global sales is around 0.5 million for each video game and the highest global sale can reach to 82.5 million.

	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Critic_Score	Critic_Count	User_Count
count	16719.000000	16719.000000	16719.000000	16719.000000	16717.000000	8137.000000	8137.000000	7590.000000
mean	0.263330	0.145032	0.077595	0.047339	0.533599	68.967679	26.360821	162.229908
std	0.813514	0.503282	0.308819	0.186709	1.548019	13.938165	18.980495	561.282326
min	0.000000	0.000000	0.000000	0.000000	0.010000	13.000000	3.000000	4.000000
25%	0.000000	0.000000	0.000000	0.000000	0.060000	60.000000	12.000000	10.000000
50%	0.080000	0.020000	0.000000	0.010000	0.170000	71.000000	21.000000	24.000000
75%	0.240000	0.110000	0.040000	0.030000	0.470000	79.000000	36.000000	81.000000
max	41.360000	28.960000	10.220000	10.570000	82.530000	98.000000	113.000000	10665.000000

Section 5 Data Exploration

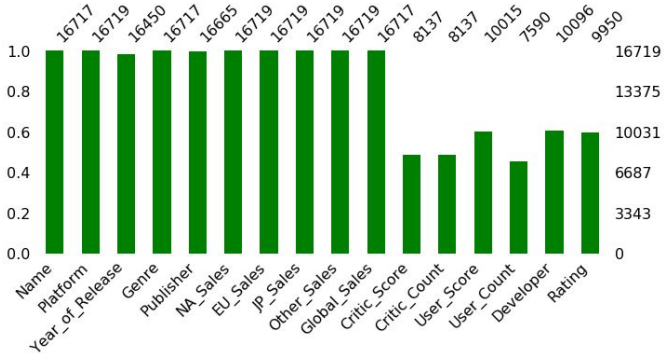
Here we show some visualization result of the data from different perspectives. The objective of doing this is to explore the characteristics of the data and gain more insights into the data. And this greatly facilitates our selection of features and models.

1) Correlation Matrix of variables



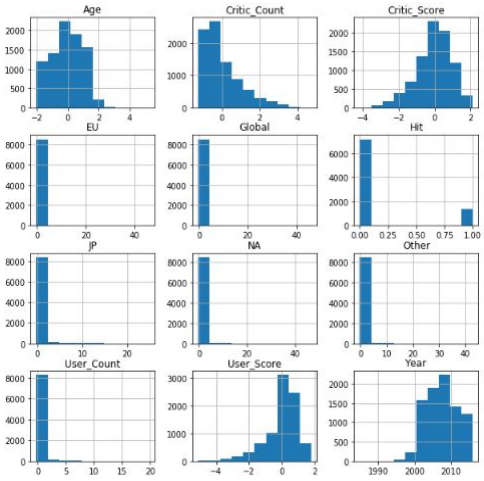
First, we investigate the relationship across different categories of sales. As displayed in the Pearson Correlation coefficient matrix, there is a strong correlation between regional sales, which includes JP, EU and NA, and global sales.

2) Missing value of the dataset



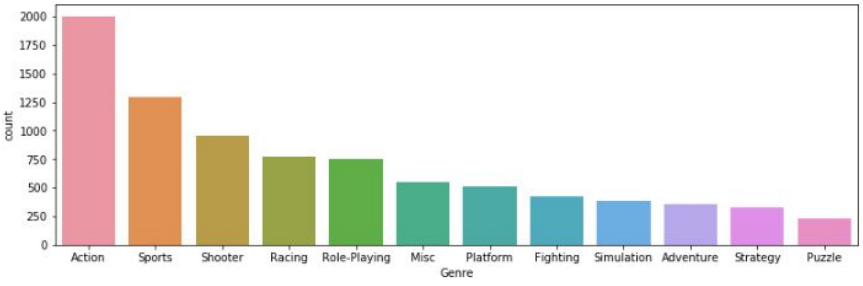
Next, we calculate the missing values for each column. The above bar chart shows that missing values is very serious in this dataset, there are around 50% of data are missing in some columns. Among them, Critic_Score, Critic_Count, and User_Count have the largest amount of missing values, which have 51.5%, 51.5% and 54.6% missing ratio, respectively.

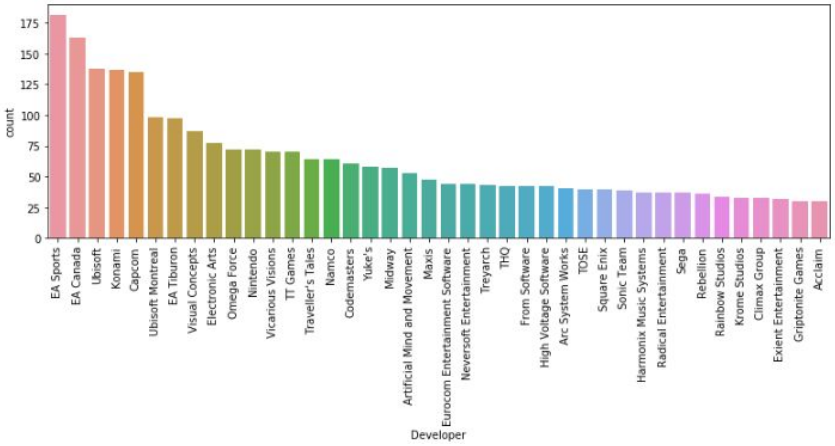
3) Variables Distribution

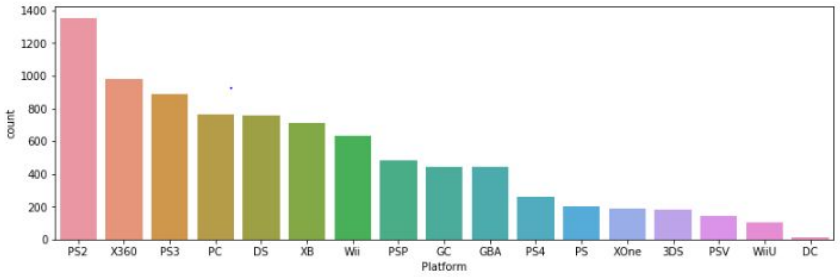


Then, we would like to plot the histogram graph to see the variables distribution within its value range. For example, from the distribution of 'Year' (Year of release), it shows that after 2000, there is a sharp increase afterwards, which is associated with the boost of the game market. In addition, as Critic_Count, User_Count, and User Score are highly skewed, thus we need to consider transform these variables in the later.

4) Categorical variables

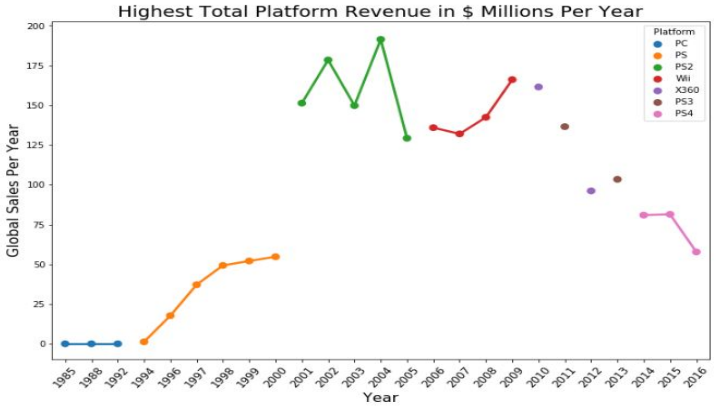






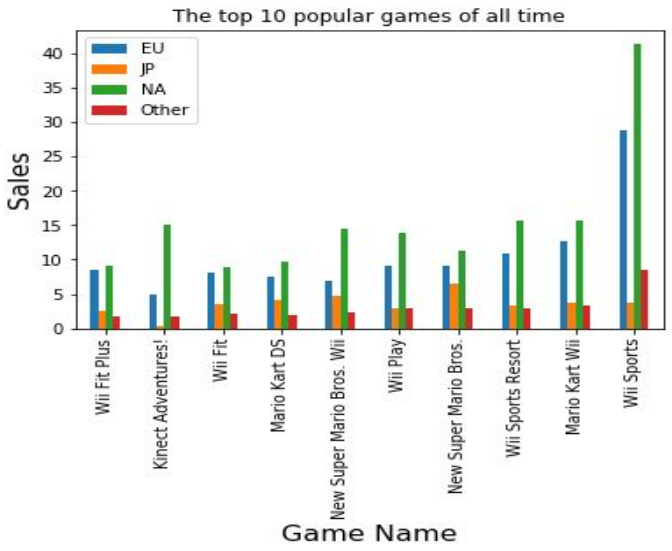
As the previous graph is only limited to the numeric variables, thus we are interested in exploring the distribution of categorical variables by plotting them separately. The above graph indicates the distribution of 'Platform', 'Developer' and 'Genre'. We can see the 'PS2', 'X360' and 'PS3' are ranked as top 3 popular platforms. 'EA sports' is the most productive developer among all developers. In addition, 'Action', 'Sports', 'Racing' and 'Shooter' contributes to the majority of game Genre.

5) Highest total platform revenue in $ millions per year



This graph illustrates the highest platform revenue in dollar millions ranging from 1985 to 2016. Here, the x-axis represents the Year (which is year of game release before renaming). As is shown, the different platform takes lead in different times, and it seems like Sony PlayStation series (PS, PS2, PS3, PS4) earns the most popular throughout these years.

6) The top 10 popular video games of all time

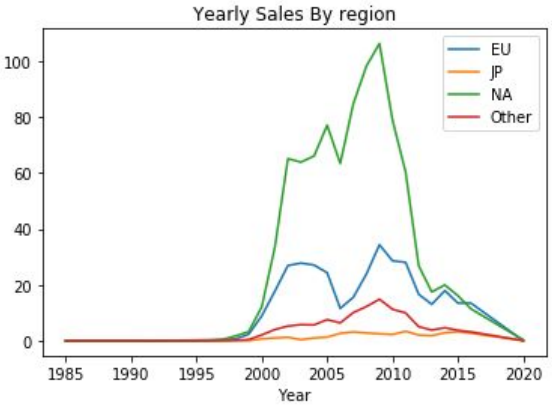


It is obvious to see that the trend of the most popular video game is similar across different regions because of the high correlation between them. However, some games (e.g. Kinect Adventures) are less popular in JP, while very popular in NA. Also, 'Wii Sports' won the largest market share in all areas except JP.

7) The top 10 publishers by sales

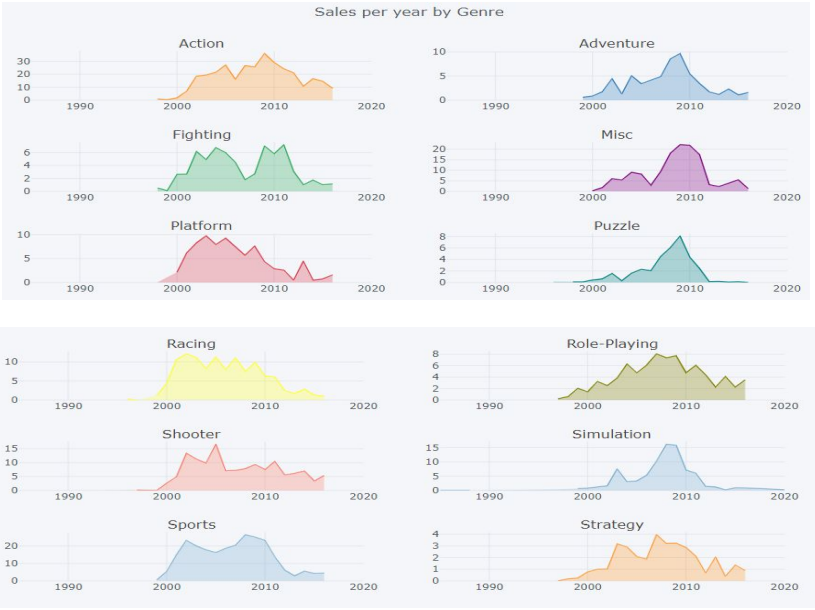
In addition, according to the relationship between publisher and sales, we can build a sense of which publishers are more popularity. In North America and Europe regions, 'Electronic Arts', 'Activision' and 'Nintendo' won the battle. In contrast, in the Japanese market, 'Sony Computer' and 'Nintendo' are in the leading place, which matches the popularity of PlayStation series and Nintendo in the region as well.

8) Yearly Sale by Regions



Here, we draw the yearly sales by different region. As is shown in the figure, since 2000, the sales in the different region started to go up and the spike happened around 2010. It shows the fast development of video game in recent years.

9) Sales per Year by Genre



For the different game genre, we analyzed the corresponding yearly sales. As is shown in the above figures, we can confirm the consistency across different genres. Most of them have peak sales around 2010 even though their trends follow various patterns.

10) Comparison of the sales of PS2, PS, PC, X360, Wii, PS3, PS4 from the year 2000 to 2016

Finally, this matrix helps us to explore the relationship between regional sales and platforms. For example, compared to Japan, Europe, and North America contributed more revenues to PS4, PS3, Wii, PS and X360 platform.

Section 6 Data Prepossessing

	Missing Values	% of Total Values
User_Count	9129	54.6
Critic_Score	8582	51.3
Critic_Count	8582	51.3
Rating	6769	40.5
User_Score	6704	40.1
Developer	6623	39.6
Year_of_Release	269	1.6
Publisher	54	0.3
Name	2	0.0
Genre	2	0.0
Global_Sales	2	0.0

Data preprocessing is an essential data mining step to transform raw data into a more understandable way and thus can prepare for the data modeling. Specifically, we have utilized data cleaning, data modification, and data reduction when preprocessing the video games dataset.

This dataset, as mentioned, suffers the serious problem of missing value. About 50% of the data are incomplete cases and have no information in the 6 columns including ‘User_Count’, ‘Critic_Score’, ‘Critic_Count’, ‘Rating’, ‘User_Score’ and ‘Developer’. However, they might be essential for the predictions, so we can’t simply delete these columns entirely. We need to find an appropriate strategy to deal with the missing value at first and then make the necessary changes to restructure the current dataset.

Methodology：

1. Drop records without the year of release or Genre.
2. There are 2 records with the year of release “Adventure”-they are obviously wrong values for the year. Thus, we deleted those 2 rows.

3. In “User_Score” column, there are values "tbd", and we marked them as NaN.
4. For many rows in the data frame, all 6 columns including Rating, Developer, User_Count, User_Score, Critic_Count, and Critic_Score are missing. In order to improve the model precision, we deleted those rows with all last 6 columns are blank.
5. Impute the missing value for the following columns: User_Count, User_Score, Critic_Count, and Critic_Score. We replaced the missing value with the median of that column respectively.
6. Use the log algorithm to transform “Critic_Count”, “User_Count”, and “User Score” from highly skewed variables to the normal distribution.
7. Create dummy variables for categorical variables and only keep the dummy variables with high frequency for “Developer” and “Publisher”.
8. Create a new binary column ‘Hit’ based on ‘Global_sales’. If the Global_sales’ is greater than 1 million dollars, the Hit is 1, which means that video game is classified as popular; otherwise, the ‘Ht’ is 0.
9. Correlation matrix for numeric variables shows there is a strong correlation between regional sales and global sales. In this dataset, the global sale is actually the sum of all regional sales. To increase the model precision, we just keep the global sales only as the target in further prediction and remove the rest sales, including ‘JP_sales’, ‘NA_sales’, ‘EU_sales’ and ‘Other_sales’ to decrease the prediction bias.

Section 7 Build Models

According to our previous explanation, we are now having two separate tasks, the first one is making the prediction of the number of global sales for each video game (regression), another one is making the prediction of whether a video game is popular (classification). To begin with, we build several models by utilizing different machine learning algorithms. Next, we will select the best models for each task based on the model comparison. From then on, we would like to move to the regression task firstly, and later we will discuss the classification task.

Considering the regression, the target variable is determined as the global sales of video games. After the above data preprocessing stage, we retain 59 variables total with dummy variables. Before building models, in order to enable the machine learning algorithm to train faster, improve the accuracy of a model by choosing the right subset, reduce the complexity of a model, and avoid the overfitting issue, we choose to use feature selection method by setting the k equal to 24, which indicate select the 24 top features. In addition, we split the data into training data and testing data: training data is used to train the model using the corresponding algorithm, while testing data is used to evaluate the model.

```
# Create and fit selector
selector = SelectKBest(f_classif, k=24)
selector.fit(X, Y)
# Get idxs of columns to keep
mask = selector.get_support()
new_features = X.columns[mask]
new_features
```

Part 1： Regression Task

Because we make the prediction of continuous variables in this particular task, therefore, we build models with five machine learning algorithms that well fit for regression, including Linear Regression, Random Forest, k Nearest Neighbor (kNN), Neural Network and Gradient Boosting.

- **Linear Regression**

```
from sklearn.linear_model import LinearRegression
lm=LinearRegression()
lm.fit(X_train,Y_train)

pre_train=lm.predict(X_train)
pre_test=lm.predict(X_test)
```

Linear regression model can act as a baseline model and allow as to see if the independent variables have a linear relationship with dependent variables.

- **Random Forest**

```
# Import the model we are using
from sklearn.ensemble import RandomForestRegressor
# Instantiate model with 1000 decision trees
rf = RandomForestRegressor(n_estimators = 1000, random_state = 42)
# Train the model on training data
rf.fit(X_train, Y_train)
```

- **Gradient Boosting**

```
from sklearn.ensemble import GradientBoostingClassifier
model= GradientBoostingRegressor(min_samples_leaf= 4, learning_rate= 0.1, max_depth= 4)
# Train the model using the training sets and check score
model.fit(X_train, Y_train)
#Predict Output
predictions= model.predict(X_test)
```

Gradient Boosting build trees one at a time, where each new tree helps to correct errors made by the previously trained trees, whereas Random Forest trains each tree independently, using a random sample of the data. We set the number of trees in Random Forest as 1000 and adjust the minimum leaves and maximum depth for Gradient Boosting in order to increase their prediction power.

- **kNN**

```
rmse_val = [] #to store rmse values for different k
for K in range(10):
    K = K+1
    model = neighbors.KNeighborsRegressor(n_neighbors = K)

    model.fit(X_train, Y_train)  #fit the model
    pred=model.predict(X_test) #make prediction on test set
```

- **Neural Network**

```
# Adding the input layer and the first hidden layer
model.add(Dense(32, activation = 'relu', input_dim = 59))
# Adding the second hidden layer
model.add(Dense(units = 32, activation = 'relu'))
# Adding the third hidden layer
model.add(Dense(units = 32, activation = 'relu'))
# Adding the output layer
model.add(Dense(units = 1))
```

```
model.compile(optimizer = 'adam',loss = 'mean_squared_error')
```

```
model.fit(X_train, Y_train, batch_size = 10, epochs = 10)
```

```
predictions = model.predict(X_test)
```

For kNN, we try different k values ranging from 1 to 10 and compare the testing result to choose the optimal k size. For Neural Network, our model includes one input layer, three hidden layers, and one output layer. We use ReLU as activation function and let the model run ten iterations to train the parameters.

Part 2： Classification Task

As we aim to predict whether the video game will be popular the in market in the future, thus our target for classification task would be the column 'Hit'. Accordingly, we built four models using support vector machine (SVM), logistic regression, decision tree, and Gaussian Naive Bayes algorithms.

- **Support Vector Machine**

```
svclassifier = SVC(kernel='linear')
svclassifier.fit(X_train, Y_train)
predicted = svclassifier.predict(X_test)
```

- **Logistic Regression**

```
lgmodel = LogisticRegression()
# Train the model using the training sets and check score
lgmodel.fit(X_train, Y_train)
lgmodel.score(X_train, Y_train)
```

- **Decision Tree**

```
dtmodel = tree.DecisionTreeClassifier(criterion='entropy', max_depth=3, random_state=0)

dt=dtmodel.fit(X_train, Y_train)
dt.score(X_train, Y_train)
```

- **Gaussian Naive Bayes**

```
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(X_train, Y_train)
```

For SVM, we select linear kernel as it performs much faster compared to other types of kernels (e.g. polynomial kernel) for our data. For decision tree, we use entropy as the criterion to calculate the homogeneity of a sample and set max depth to 3. Logistic regression and Gaussian Naive Bayes are pretty straightforward by using default parameters as they are commonly used for classification prediction.

Section 8 Model evaluation and selection

- **Performance of different Regression models**

	RMSE	**MAE**	**Explained variance score**	**R2**
Linear regression	**0.0570**	**0.0302**	**0.2632**	**0.2626**
Random forest	**0.00853**	**0.0497**	**0.8238**	**0.9536**
kNN	**0.00895**	**0.0253**	**0.9856**	**0.9850**
Gradient Boosting	**0.00088**	**0.0177**	**0.7994**	**0.7992**
Neural Network	**0.0126**	**0.0896**	**0.2279**	**0.2366**

After running all the models, we are able to compare the model performance across different regression models. To properly evaluate the models, the above table presents the accuracy assessment for all regression models by including the four selection criteria.

(1) Mean square error (MSE)

(2) Explained variance score

(3) R squared (R2)

(4) Root mean square error (RMSE)

One thing worth mentioning is that kNN model achieves the highest R squared score of 0.985. However, since many variables in our dataset are binary variables, KNN algorithm actually does not perform well as we have eventually found that the KNN model has clustered each observation separately with k=1. This is because it is difficult to find the distance between dimensions with categorical features. As a result, we do not consider kNN as a good candidate for regression task.

To sum up, Gradient Boosting model and Random forest model are our best models. The

Gradient Boosting model has the lowest root mean square error which indicates the highest model accuracy, and the Random Forest has highest R square among all the models.

- **Performance of different Classification models**

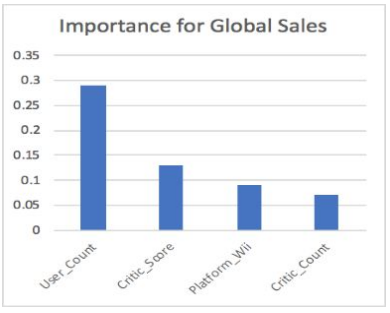
	Accuracy score	Recall score	Precision score	AUROC
Decision Tree	**0.7739**	**0.8008**	**0.7654**	**0.7734**
Logistic Regression	**0.7667**	**0.7602**	**0.7709**	**0.7667**
SVM	**0.7343**	**0.6957**	**0.7582**	**0.7347**
Gaussian Naïve Bayes	**0.5804**	**0.4741**	**0.5246**	**0.5110**

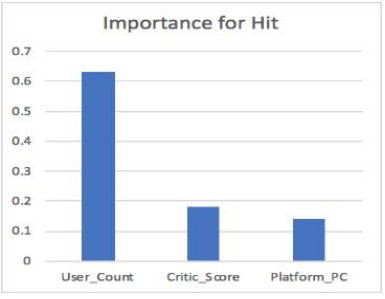
When moving to classification models, we use different sets of metrics to evaluate the accuracy of classification, including accuracy score, recall score, precision score and area under the receiver operating characteristic curve (AUROC).

In our opinions, AUROC value is the most important factor when comparing different models as it can show the model diagnostic ability while classifying a binary system.

To sum up, by looking at other criteria as well, we can conclude decision tree outperforms other models as it achieves the highest AUROC value, as well as having good accuracy scores and recall score.

- **Features Findings**





The building model is not only able to show the prediction accuracy but can also let us know the predictors' importance when making the prediction. For the regression task, the important variables are 'user_count', 'critic_score', 'Platform_Wii' and 'Critic_count'. For classification task, the important variables for are 'user_count', 'critic_score' and 'Platform_PC'.

In conclusion, the number of users who submit the reviews in Metacritic plays an important role in both tasks and it even acts as a decisive role of the popularity prediction because the percentage of importance is over 60%. The other mentioned factors are also quite important and worth being noticed.

Section 9 Conclusion and Implication

Based on our previous findings, we would like to transfer the analysis into business insights and make a couple of business recommendations to the relevant companies.

- Focus on the platform of video games like the video games on Wii platform tend to have the higher amount of business sales whereas the games in PC platform tend to be more popular.
- The number of users who give the score at Metacritic and the aggregate score compiled by Metacritic's staff can indicate whether the market prospect is positive for that video game.
- Decrease the weight of the game genre when considering the game popularity and the sales from the business perspective.

In addition, to utilize the best models for both global sales and popularity, the models can be employed in different business scenarios.

Regarding the global sales, the gradient boosting model has the lowest root mean square error which indicates the highest accuracy among all the models. Thus, this model can be used by the marketing or sales department when developing the years' sales strategy or marketing plan, it can help the company reduce its business loss. While, as the random forest model appear as the best model to fit the dataset with highest R square and thus can be used in internal statistical research when the business would like to train the model with additional factors.

Regarding the Popularity Hit variable, we would like to recommend the business employ the decision tree model for classification as it has the highest accuracy. Also, the result that derives from the model can be treated as video games guidance for both particular countries and regions. One thing needs to pay attention is that the accuracy of the popularity prediction is lower than what we have expected, which could indicate some additional factors need to be considered but are not in the dataset, such as the investment have been made for each video game, the music quality, and etc. Including those factors may potentially increase the prediction accuracy of popularity

Section 10 References

- https://www.kaggle.com/rush4ratio/video-game-sales-with-ratings/data
- https://www.ibisworld.com/industry-trends/market-research-reports/information/motion-picture-sound-recording-industries/video-games.html
- https://stackabuse.com/implementing-svm-and-kernel-svm-with-pythons-scikit-learn/
- https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
- https://datascienceplus.com/extreme-gradient-boosting-with-python/
- https://stackoverflow.com/questions/49008074/how-to-create-a-neural-network-with-regression-model
- https://towardsdatascience.com/random-forest-in-python-24d0893d51c0
- https://stackabuse.com/k-nearest-neighbors-algorithm-in-python-and-scikit-learn/