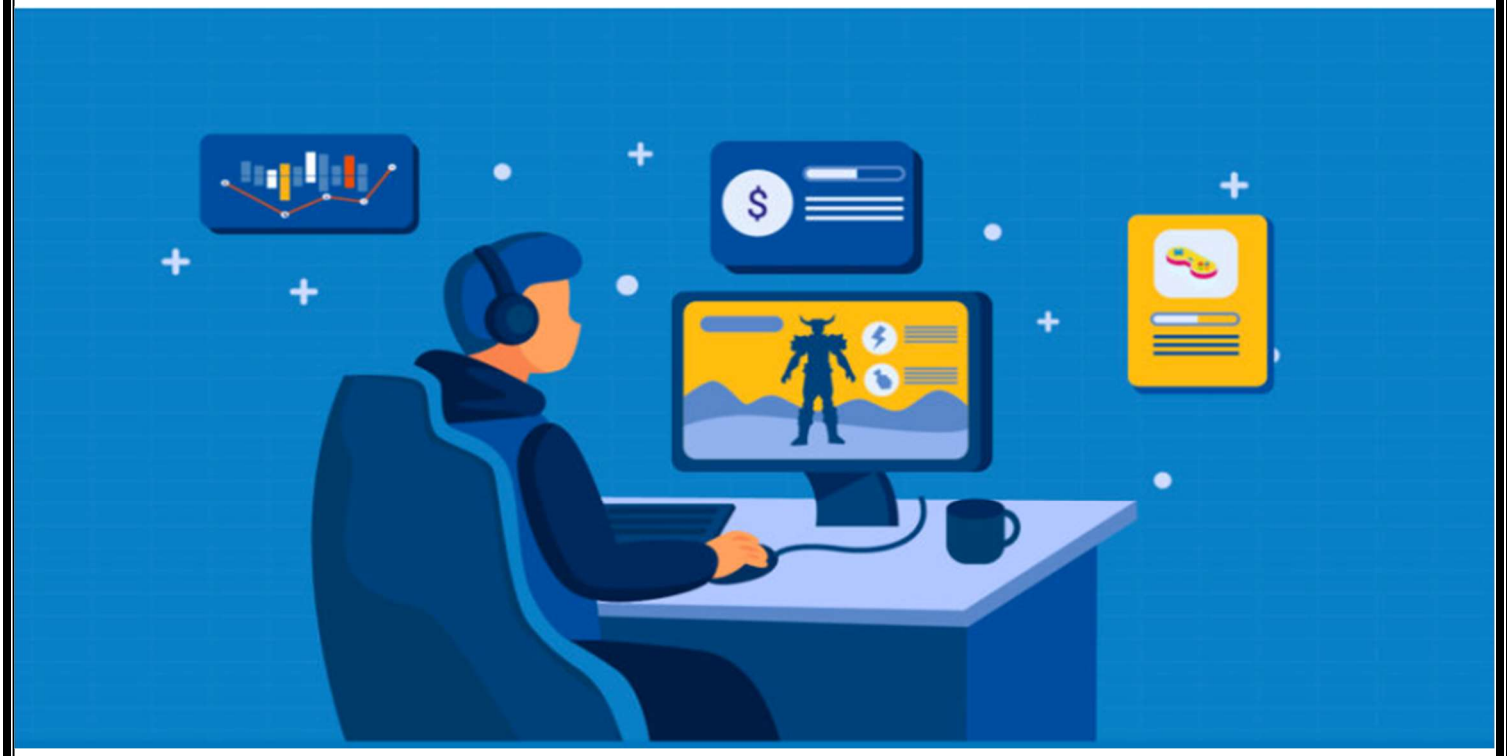


Game Success Classification

PROJECT REPORT

Submitted By

Allika Thadishetty

Genchen Li

Sneha Chandrashekar

Sreerag Mahadevan Cheeroth

Tuan Nguyen

Xinyue Li

BUDT 704 GROUP 10

BUDT 704

GAME SUCCESS CLASSIFICATION

ABSTRACT

Video games have been one of the fastest-growing industries in the last decade, with revenues increasing from 32 billion dollars in 2012 to 192 billion dollars in 2022. In 2021, Steam recorded a total concurrent player count of 26.09 million, and this figure only applies to players in their platform games. In the long run, having such a large amount of traffic provides any game developer with enough and more data on user behavior to help predict the success of specific games. In this project, we attempt to correlate a few independent variables in order to successfully predict the overall success rate of particular games in a given sales region.

INTRODUCTION

Microsoft currently has a monopoly on many of the major game franchises, which is expected to generate over 180 billion dollars in revenue by 2022. In today's markets, it's difficult for a game developer to track the player base's trends in order to better predict the success rate of specific games in a given region. There are numerous factors that could influence the decision, including the genre, ESRB rating, and overall success of the publisher's previous projects, and having a fundamental model that accepts a set of independent variables to better predict the sales figure would aid in determining the genre of their next product.

Every year, the gaming industry produces extremely impressive figures in terms of earnings and concurrent player counts. With the figures always being in the millions to billions of dollars, any critical information that can provide a lead to a game developer would be instrumental in reaping profits within their predetermined risk range.

The project is subdivided into the following sections. These are:

- Data collection
- Data Preprocessing
- Data Visualisation
- Model Selection
- Transformation of Variables
- Training and testing datasets.
- Results and Performance
- Conclusion

DATA

Our Dataset is from Kaggle Database. It contains games and a lot of other detailed information. It contains 55792 games in it with 21 columns of information about the games. Since we are trying to analyze and predict sales of the games and how sales will change while the region changes,and then predict the maximum sales of games, we first remove all the irrelevant columns from the dataset, and then we move to the data cleaning process, which includes removing null values and also adding a new column called country_max.

Dataset URL: https://www.kaggle.com/datasets/gregorut/videogamesales

Columns: Rank, Name, basename, Genre, ESRB_Rating, Platform, Publisher, Developer, VGChartz_Score, Critic_Score, User_Score, Total_Shipped, Global_Sales, NA_Sales, PAL_Sales, JP_Sales, Other_Sales, Last_Update, status.

```
data = pd.read_excel("D:\\python\\Big_Data.xlsx")
data = pd.DataFrame(data)
```

```
data.head()
```

	Rank	Name	basename	Genre	ESRB_Rating	Platform	Publisher	Developer	VGChartz_Score	Critic_Score	...	Total_Shipped	Global_Sales	NA_Sales	PAL_Sales	JP_Sales	Other_Sales	Year	Last_Update	Unnamed: 19	status
0	1	Wii Sports	wii-sports	Sports	E	Wii	Nintendo	Nintendo EAD	NaN	7.7	...	82.86	NaN	NaN	NaN	NaN	NaN	2006.0	NaN	NaN	1
1	2	Super Mario Bros.	super-mario-bros	Platform	NaN	NES	Nintendo	Nintendo EAD	NaN	10.0	...	40.24	NaN	NaN	NaN	NaN	NaN	1985.0	NaN	NaN	1
2	3	Mario Kart Wii	mario-kart-wii	Racing	E	Wii	Nintendo	Nintendo EAD	NaN	8.2	...	37.14	NaN	NaN	NaN	NaN	NaN	2008.0	11th Apr 18	NaN	1
3	4	PlayerUnknown's Battlegrounds	playerunknowns-battlegrounds	Shooter	NaN	PC	PUBG Corporation	PUBG Corporation	NaN	NaN	...	36.60	NaN	NaN	NaN	NaN	NaN	2017.0	13th Nov 18	NaN	1
4	5	Wii Sports Resort	wii-sports-resort	Sports	E	Wii	Nintendo	Nintendo EAD	NaN	8.0	...	33.09	NaN	NaN	NaN	NaN	NaN	2009.0	NaN	NaN	1

5 rows × 21 columns

This is a multi-class classification problem.

METHODS

- **Data Pre-processing Steps**

Step 1: Removing all the irrelevant columns from the dataset

We chose to drop the 'Rank', 'Name', 'basename' 'VGChartz_Score', 'Global_Sales', 'Other_Sales', 'Unnamed:19', 'Last_Update', 'Status' columns.

```
data = data.drop(['Rank','Name','basename','VGChartz_Score','Global_Sales','Other_Sales','Unnamed: 19','Last_Update','status'],axis=1)
```

Step 2: Cleaning the Dataset.

1. Find and remove all the null values from 'NA_Sales', 'PAL_Sales', and 'JP_Sales'.

```
#finding null values
print("NULL VALUES IN")
print("PAL :",data['PAL_Sales'].isna().sum())
print("NA :",data['NA_Sales'].isna().sum())
print("JAPAN :",data['JP_Sales'].isna().sum())
```

```
NULL VALUES IN
PAL : 42603
NA : 42828
JAPAN : 48749
```

```
#removing the null values
data = data[data['PAL_Sales'].notna()]
data = data[data['NA_Sales'].notna()]
data = data[data['JP_Sales'].notna()]
#data = data.reset_index(drop=True, inplace=True)
```

	Genre	ESRB_Rating	Platform	Publisher	Developer	Critic_Score	User_Score	Total_Shipped	NA_Sales	PAL_Sales	JP_Sales	Year
19	Action	M	PS3	Rockstar Games	Rockstar North	9.4	NaN	NaN	6.37	9.85	0.99	2013.0
20	Action	M	PS4	Rockstar Games	Rockstar North	9.7	NaN	NaN	6.06	9.71	0.60	2014.0
30	Action	M	PS2	Rockstar Games	Rockstar North	9.6	NaN	NaN	8.41	5.49	0.47	2002.0
32	Action	M	X360	Rockstar Games	Rockstar North	NaN	NaN	NaN	9.06	5.33	0.06	2013.0
34	Shooter	M	PS4	Activision	Treyarch	NaN	NaN	NaN	6.18	6.05	0.41	2015.0

2. Drop 'Total_Shipped' and 'User_Score' columns from the dataset since they are not that relevant to our research, and remove all the null values from the 'Critic_Score' column.

```
#here as all the total_shipped values are null we will drop it
data = data.drop(['Total_Shipped'],axis=1)
data = data.drop(['User_Score'],axis=1)
data = data[data['Critic_Score'].notna()]
#data = data.reset_index(inplace=True)
#data = data.drop(['index'],axis=1)
data.shape
```

```
(1317, 10)
```

	Genre	ESRB_Rating	Platform	Publisher	Developer	Critic_Score	NA_Sales	PAL_Sales	JP_Sales	Year
19	Action	M	PS3	Rockstar Games	Rockstar North	9.4	6.37	9.85	0.99	2013.0
20	Action	M	PS4	Rockstar Games	Rockstar North	9.7	6.06	9.71	0.60	2014.0
30	Action	M	PS2	Rockstar Games	Rockstar North	9.6	8.41	5.49	0.47	2002.0
40	Shooter	M	X360	Activision	Infinity Ward	8.7	9.07	4.29	0.13	2011.0
41	Shooter	M	X360	Activision	Treyarch	8.8	9.76	3.73	0.11	2010.0

Step 3: Handling Data

Adding a new column named 'Country_max' and removing three original region_sale columns.

Since we are trying to analyze and predict the relationships between games' sales and regions, we decided to create a new column which is called 'Country_max'. We do a for loop here and to find out the region that has the highest game sale for each game, and we put the result into the column. After we finish that process, we then remove all the three original region_sale columns from the dataset and our dataset is now all set.

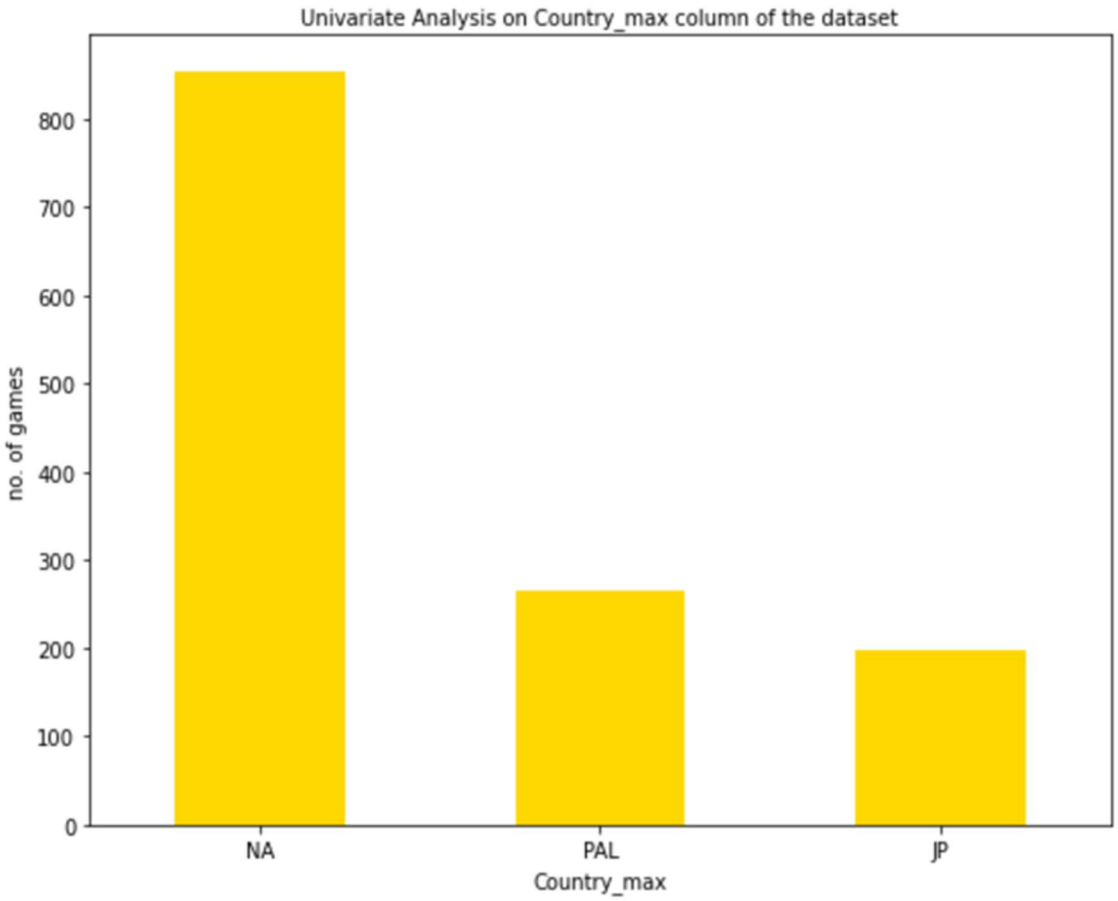
```
#finding max country column and making a new column
data['Country_max'] = "-"
for i in data.index:
    if(max(data['NA_Sales'][i],data.PAL_Sales[i],data.JP_Sales[i])== data.NA_Sales[i]):
        data['Country_max'][i] = "NA"
    elif(max(data.PAL_Sales[i],data.JP_Sales[i])== data.PAL_Sales[i]):
        data['Country_max'][i] = "PAL"
    else:
        data['Country_max'][i] = "JP"
```

	Genre	ESRB_Rating	Platform	Publisher	Developer	Critic_Score	Year	Country_max
0	Action	M	PS3	Rockstar Games	Rockstar North	9.4	2013.0	PAL
1	Action	M	PS4	Rockstar Games	Rockstar North	9.7	2014.0	PAL
2	Action	M	PS2	Rockstar Games	Rockstar North	9.6	2002.0	NA
3	Shooter	M	X360	Activision	Infinity Ward	8.7	2011.0	NA
4	Shooter	M	X360	Activision	Treyarch	8.8	2010.0	NA

➢ Data Visualisation

Plot 1:

```
#bar plot for categorical columns
plt.rcParams['figure.figsize']=(9,7)
x=data['Country_max'].value_counts()
x.plot(kind='bar', color ='gold')
plt.title('Univariate Analysis on Country_max column of the dataset', fontsize= 10)
plt.xlabel('Country_max')
plt.ylabel('no. of games')
plt.xticks(rotation = 360)
plt.show()
```



Plot 2:

```
plt.rcParams['figure.figsize']=(10,8)
sns.lineplot(data=data, x='Year', y='Critic_Score', hue='Country_max')
```

```
<AxesSubplot:xlabel='Year', ylabel='Critic_Score'>
```

➢ Data Transformation

Step 4: Label Encoding

Since most of our independent variables are categorical, it is a good practice to transform them into labels than keeping them as strings. We used LabelEncoder() in order to achieve this.

	Genre	ESRB_Rating	Platform	Publisher	Developer	Critic_Score	Year	Country_max
0	Action	M	PS3	Rockstar Games	Rockstar North	9.4	2013.0	PAL
1	Action	M	PS4	Rockstar Games	Rockstar North	9.7	2014.0	PAL
2	Action	M	PS2	Rockstar Games	Rockstar North	9.6	2002.0	NA
3	Shooter	M	X360	Activision	Infinity Ward	8.7	2011.0	NA
4	Shooter	M	X360	Activision	Treyarch	8.8	2010.0	NA

```
le = preprocessing.LabelEncoder()

data = data[cols].apply(le.fit_transform)
data.head()
```

	Genre	ESRB_Rating	Platform	Publisher	Developer	Country_max
0	0	2	13	59	306	2
1	0	2	14	59	306	2
2	0	2	12	59	306	1
3	14	2	21	7	177	1
4	14	2	21	7	382	1

```
#merging data
data = pd.concat([data_num,data], axis=1)
data.head()
```

	Critic_Score	Year	Genre	ESRB_Rating	Platform	Publisher	Developer	Country_max
0	9.4	2013.0	0	2	13	59	306	2
1	9.7	2014.0	0	2	14	59	306	2
2	9.6	2002.0	0	2	12	59	306	1
3	8.7	2011.0	14	2	21	7	177	1
4	8.8	2010.0	14	2	21	7	382	1

➢ Training Testing and Model

Step 5: Train Test Split. Keeping aside thirty percent of the data for test set, using sklearn.model_ selection

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(data.drop('Country_max',axis=1),data['Country_max'],test_size = 0.3, random_state = 7)
```

```
print("Training data shape: ",X_train.shape)
print("Testing data shape: ",X_test.shape)
```

```
Training data shape:  (921, 7)
Testing data shape:  (396, 7)
```

Step 6: Model Selection

Since this is a classification problem, where we are trying to classify which country it will be the most successful in, we decided to go ahead with the K-Nearest neighbors model commonly known as KNN.

The objective of the project was to identify the region amongst which the sales for the games would be most successful. Since our target and predictor variables are categorical, we decided to use K-Nearest Neighbor (KNN) in order to classify the data from the predictor variables (categorical) and identify the nearest 7 neighbors to identify the region of sale for the product. We took the K value as 7. If we chose a lower k value, it would have resulted in the overfitting of the model and any value for K higher would result in the underfitting of the model.

KNN is a supervised machine learning algorithm that provides us with an approach to predict and classify labeled dataset to train models to predict outcomes. In this case, would be to identify the region of sales depending on its nearest neighbor data point.

Step 7: Fitting the model on Training Data

```
from sklearn.neighbors import KNeighborsClassifier
```

```
knn1 = KNeighborsClassifier(n_neighbors = 7)
knn1.fit(X_train,y_train)
```

```
KNeighborsClassifier(n_neighbors=7)
```

```
y_pred = knn1.predict(X_test)
```

```
y_pred
```

```
array([1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 0,
       1, 1, 1, 1, 2, 2, 1, 2, 1, 1, 1, 1, 0, 0, 2, 2, 1, 0, 2, 1, 1, 0,
       1, 1, 1, 1, 1, 1, 2, 1, 1, 0, 2, 1, 0, 1, 0, 2, 1, 1, 1, 1, 1, 2,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1,
       1, 1, 1, 0, 1, 1, 1, 1, 2, 1, 0, 0, 2, 1, 1, 2, 1, 1, 1, 1, 1, 1,
       0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 1, 1, 0, 1, 2,
       2, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1,
       1, 0, 0, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 2, 1, 1, 2, 1, 1,
       1, 2, 1, 2, 0, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 0, 1, 2,
       1, 1, 1, 0, 2, 1, 2, 1, 1, 1, 2, 1, 0, 1, 1, 1, 1, 1, 1, 1, 2, 1,
       2, 1, 0, 2, 1, 1, 2, 1, 1, 2, 1, 1, 2, 1, 1, 2, 1, 1, 1, 1, 1, 0,
```

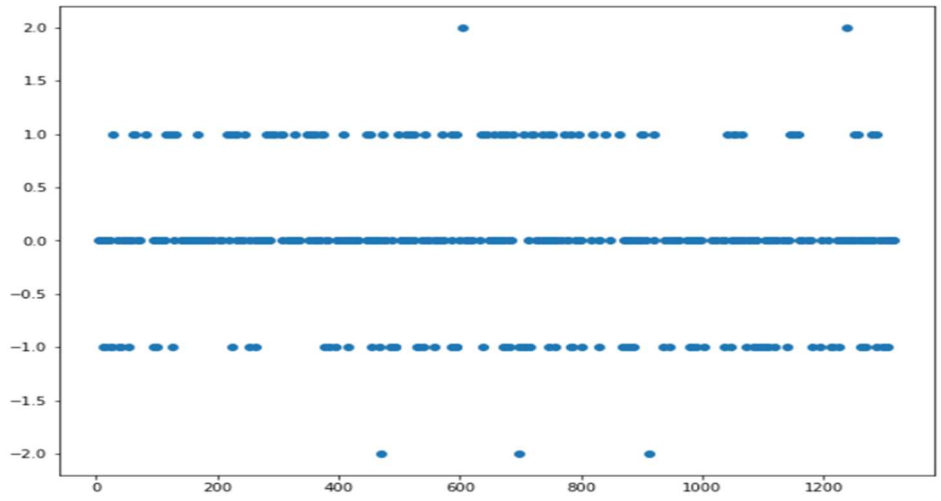
➢ Results and Performance

Step 8: Results and Performance

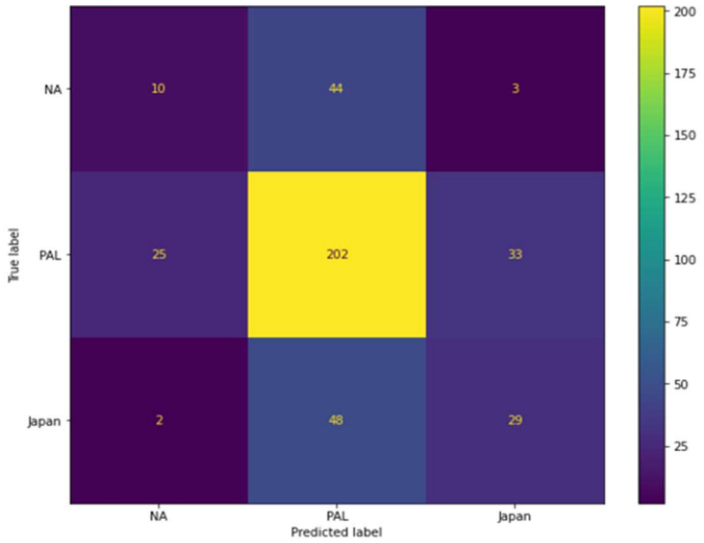
```
In [40]: print("ACCURACY SCORE :", metrics.accuracy_score(y_test,y_pred))
         RMSE=np.sqrt(np.sum(np.square(y_pred-y_test)))
         print("RMSE :", RMSE)

ACCURACY SCORE : 0.6085858585858586
RMSE : 13.038404810405298
```

```
In [41]: #PLOTTING THE PREDICTION ERROR
         g=plt.plot(y_test - y_pred,marker='o',linestyle='')
```




```
#Confusion Matrx
confusion_matrix = metrics.confusion_matrix(y_test, y_pred)
confusion_matrix
cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix = confusion_matrix, display_labels = ['NA', 'PAL', 'Japan'])
cm_display.plot()
plt.show()
```



RESULTS :

The results of our Knn model show a modest predictive power with an accuracy score above baseline accuracy of random guessing (0.6 > 0.5) with the accuracy score being calculated by dividing the total sum of True Positive and True Negative by the total number of observations. Our confusion matrix shows a clearer picture of where the model is most accurate (predicting PAL regions) and least accurate (predicting NA regions).

APPROACH:

We started with data cleaning and preprocessing, which involved cleaning missing and NaN values. Next comes identifying relevent variables and dropping the unnecessary columns. Later we transformed our variables according to the goal of the project and for easing the work with data.

We proceeded to transform our target variable from the 3 columns of Sales to a column of the name of the country with maximum sales. We used Label encoding for all our categorical predictors.

We use matplotlib for the visualisation of our data and to identify if we get an useful insights. We also used correlation matrix in order to find out relevent independent variables.

We then went ahead and selected KNN as our Classification Model, split our dataset into 7:3 as training and testing using sklearn.model_selection library.

For evaluation metrics, we used RMSE and Accuracy Score along with confusion matrix. We also plotted predicted vs actual values in order to get a better understanding of the performance of the model.

DISCUSSION:

We can improve the model in the following ways:

1. Data: the data is not hard to collect in our chosen topic (games and their performances) but data accuracy across datasets is questionable (e.g different sales figures, etc.) so we choose not to combine datasets and instead stick to only one to ensure all our data points are consistent. This has a limitation that we don't nearly have enough dimensions/features to have a robust data mining and variables selection process.

2. Descriptive and Predictive Power: our current model is quite limited in its usefulness both in descriptive analysis and predictive analysis. We would prefer a

higher accuracy score, especially with a big dataset such as the one we are using. Moving forward, we could potentially explore some kinds of trend analysis (observing how sales in each region/genre changes over time) to have more meaningful insights in our results.

3. Modeling: Knn method is easy to implement and ultimately proves to be the most accurate in our quest to have any degree of predictive power in the dataset that we chose to work on. This method is also very accommodative if we do decide to integrate new dimensions to our dataset for better predictive power.

4. Preliminary analysis: we initially try to analyze from the angle of a game developer i.e assessing how efficient companies are translating their inputs (measured in terms of game size, costs, languages and downloadable contents) into output (number of followers). We used Data Envelopment Analysis (DEA) and came up with an interesting observation with efficiency bias toward smaller games. Then we ran a regression with rating as dependent variable and game efficiency ratio as independent variable, and we found that the game efficiency ratio has a negative impact on the rating, which means based on our model, to some extent, a high game development efficiency might not ensure a high rating. However, we ended up giving up this direction as later data proved inconclusive.

REFERENCES

1. Game Sales dataset, https://www.kaggle.com/datasets/gregorut/videogamesales

2. Project Report, https://www.academia.edu/43874328/Introduction_to_Data_Science_PROJECT_REPORT_Submitted_By

3. Model Selection, https://scikitlearn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html