Predicting And Analysing Game Sales, Ratings, and Reviews On different Platforms Using Machine Learning Models

d

Ibrahim Rinub Babu
National College of Ireland
Dublin, Ireland.
x19207387@student.ncirl.ie

Abstract- ***This paper explains the implementation of machine learning algorithms by training the model with the datasets which have been cleaned, transformed, pre-processed, analyzed, and also normalized concerning the algorithms. Six models are implemented with different classic machine learning algorithms such as KNN (K-Nearest Neighbours) and SVM (Support Vector Machine) and some tree-based algorithms such as Random Forest and Decision tree. All the three datasets used are related to Games on steam and game sales on different platforms such as Pc, Xbox, and Playstation, and also Apple app store Strategy Games.***

For the imputation of missing values in the dataset, algorithms such as Mice and Missing Forest has been implemented. Algorithms such as SVM, KNN, and Random forest Classifier, and Random forest regression are built to interpret several questions and evaluated how accurately the algorithm provides the information using several factors. Logistic regression is used in Dataset 2 to predict the reviews of the Games in the Apple app store

In Dataset 3, the Decision tree and the RIPPER Rule are used to build the prediction model for predicting some of the important factors in the Dataset. The decision tree expanded to 176 branches when applied to the Steam games data collection, while the RIPPER Rule is applied for the same dependent and independent variables. The goodput is taken into account during the review process to determine exact project resources. When compared to common approaches in the literature, experimental findings show that the proposed ML-based prediction shows promising performance in terms of load prediction and for this dataset's

Keywords- Random forest, KNN, SVM, Logistic Regression, Datamining, Factorial Analysis, Data transformation, Building model

I. INTRODUCTION

The majority of the research admits that the video game market will ascend to 90 billion dollars in the upcoming years. Whereas both major and minor gaming studios are as of now running trials with utilizing such imaginative innovations as AI, VR, Blockchain, and others, within the closest future. we may anticipate something truly astonishing. By using appropriate machine learning algorithms, the forecast of global sales turnovers, mental effects of the users, and game ratings can be achieved and used for the developers to construct the game accordingly. This machine learning model can bring some changes in the game design pattern and improves the success rate, reduces the critic score, and improves the awareness of mental health while designing the game. This dataset is obtained from Kaggle. It consists of 16 columns and 16,720 rows. This data is web scraped by Gregory Smith from the website of VGChartz Video Games Sales. Some of the variables are web scraped from Metacritic and merged to prepare the final dataset. This dataset is obtained from the Kaggle. This dataset includes a compilation of video games that have sold over 100,000 copies. A scrape of vgchartz.com was used to build it. This one was selected for the project because it had a large number of samples and a combination of nominal, ordinal, and numerical variables. This had a lot of factors to fit into the machine learning model

The most important research question for this dataset was:

1. How accurately can we predict the Rating of the games using several factors and genera of the game?

In contrast, as part of the inquiry, the following secondary research questions about implementation were addressed.

2. Can the forecast of the Global Sales is done accurately using different factors and genera of the game?

3. How much can the accuracy of the prediction be improved by fitting with different algorithms and by performing the feature analysis?

The digital media industry is extremely profitable, with businesses make bulk investing huge in the production and promotion of these games to audiences. This dataset holds information about numerous games and their reviews which is extracted from the Apple app store. This set of data is being used to gain some insight into factors affecting the game review and sales and the number of downloads from this platform of this market. This is a bold move, as Apple has previously concentrated on exclusive games. Some Apple Arcade games provide continuous updates to popular App Store games. Such games can undoubtedly aid in the reduction of churn. Some people simply enjoy playing chess repeatedly. So, they could start paying an Apple Arcade membership just to keep using the same application after subscribing to play some games. This dataset is obtained from the Kaggle to perform data mining. It has numerous factors that can be used for the analysis and building machine learning model. Some of the factors such as game pricing, rating, review, and some information about the game review, size of the application This one was selected for the project because it had a large number of samples and a combination of nominal, ordinal, and

numerical variables. This had a lot of factors to fit into the machine learning model

The main research question for this dataset was:

1. Does the review of the game is flawlessly influenced by numerous factors such as language, price, size of the game, and age restriction?

In contrast, as part of the inquiry, the following secondary research questions about implementation were addressed.

2. How much can the accuracy of the prediction be improved by fitting with different variables and by performing the feature analysis to identify the variables that should be used?

Steam is the most common digital distribution site for PC games, accounting for roughly 75% of the market in 2013. By 2017, Steam users had spent approximately usd4.3 billion on games, accounting for at least 18% of global PC game revenue. They provide a lot of software for different platforms and API is known as Steamworks by using these game developers can incorporate many functions into their application as plugins or an extension. They also developed mobile apps which run on both iOS, Windows, and Android. They also produce game soundtracks and film soundtracks and also anime soundtracks.

The main research question for this dataset was:

1. How much accuracy can be achieved in forecasting the game's success and the number of owners by average ratings of these games were taken into account?

In contrast, as part of the inquiry, the following secondary research questions about implementation were addressed.

2. How much can the accuracy of the prediction be improved by fitting with different algorithms and by performing the feature analysis?

II. RELATED WORK

This section addresses journal articles and conference papers that are important to the previous section's research questions.

2.1 Dataset-1 – Video Game copy sales and rating

Whereas this report aims to apply a very simple classification algorithm to the dataset, numerous studies have been conducted to cast doubt on the suitable numerical methods. In the paper [5] Random Forest algorithm and its process of bagging aggregation from multiple trees are explained clearly. This helped to fit the dataset perfectly to the machine learning model and to tune it to achieve improved accuracy. The paper [1] discusses how to develop an analytic tool that will provide them with the information they need to deliver a personalized customer experience. The Random Forest Classification, a machine learning algorithm that is been used in dataset 1, was thoroughly investigated. This and other classification algorithms will aid us in gaining a deeper understanding of our customers and in developing marketing and communication strategies. The paper [17] briefly presents some of the core structures of the video games framework. And it primarily provides additional optimization for connect6's current computer games framework. In the evaluation function, the machine learning algorithm will improve the dynamic width adjustment in the MTD (f) framework and the self-learning model by the TD BP algorithm, which can enhance the "thinking" ability on the digital game system in connect6. It gives some ideas to develop this project in the future to the next level. The importance of the feature selection component for data mining, machine learning, and pattern recognition is explained in this article [7]. Also, how distance is important in the theory of Support Vector Machines (SVM). While the Relief-F optimization algorithm resolves function inconsistency, it does not guarantee the full distance. It also explains how to solve the problem by proposing a feature subset selection algorithm that uses SVM average distance as that of the approximation rule and serial forwarding selection as the search strategy. Paper [16] gives a detailed description of building a KNN model for prediction. The paper [13] explains the development of optimal learning algorithms from gameplay.

2.2 Dataset-2 – Apple app store games review and rating

In Dataset 2, Logistic regression is implemented to build a cost function, resolve problems, and optimize the process in the state of regression and classification problems. One paper [2] also clearly explains how logistic regression performs by using gradient descent methods to resolve optimal parameters of missing function and how Swarm intelligence algorithm such as artificial fish swarm can partially substitute the conventional penalty function approach in ensuring the optimization algorithm's global convergence. The paper [20] clearly explains the building of the logistic regression model to predict customer satisfaction. The paper [12] elaborates how the implementation of mathematical normalization aspect to the logistic regression optimization problem, and it primarily includes a significant parameter called the regularization factor that must be calculated. In paper [19] DNA sequence is analyzed, and model is built using SVM and decision tree

2.3 Dataset-3 – Stream store game price and ratings

In dataset 3, models such as the Decision tree and Ripper rule model are built for predicting the game's success and the number of owners by average ratings. In the paper [5] The feasibility of using Decision Tree Algorithms such as C4.5 Decision Tree, bagging ensemble meta-algorithm, and Random Forest algorithm is the subject of this paper's study. The Decision Tree Algorithm serves as a foundation for data classification, and begging will improve the algorithm's consistency and accuracy. The paper [8] explains how the social data had been analyzed using the SAS Enterprise Miner software tool, especially the regression and decision tree models. The outcome of the study shows that the decision tree model produced higher variable selection than the prediction model in predicting whether an individual is likely to live in the south, at least for the data set we used. The working of RIPPER, a JRip data mining algorithm, is explained in this article[6]. The JRip algorithm, which is a WEKA implementation of the RIPPER algorithm, was trained using a training dataset of 6000 URLs. The training dataset will create a model that will be used to predict the 1050 URLs in the research dataset. Finally, accuracy is assessed, and the RIPPER algorithm's success is clarified. For the prediction of imbalanced Software Defect data, this paper [4] proposes EMR SD, an Ensemble MultiBoost based on the RIPPER classifier. To begin, the algorithm employs the principal component analysis (PCA) approach to extract

the most effective features from the data set's original features, to reduce the dimension of the data, and eliminate heterogeneity.

IV. METHODOLOGY

On all three datasets, this paper used a KDD technique for data mining.

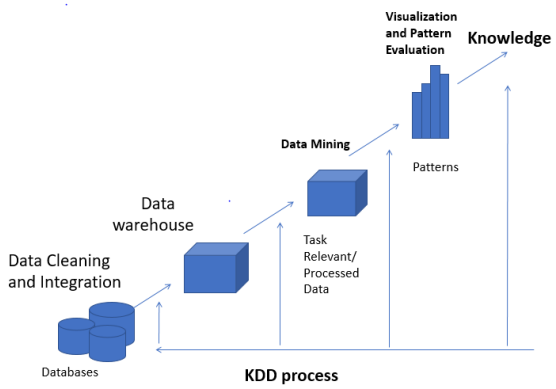

Figure 1: KDD Methodology

All three datasets follow the same sequence of steps shown in Fig 1 but are just different methodologies and different machine learning algorithms.

1.All the tree datasets were imported to the data frame and dependent variables are selected concerning the research question that we frame.

2.All the unwanted scrap variables are dropped from the dataset

3.Checking for the number of null values and deciding the algorithms such as mice, missing forest to impute, or imputing it with mean, median, mode.

4.After imputation, the dataset is analyzed on different parameters such as outliers and correct values or transforms to desired data types.

5. Identifying the outliers using boxplot, quartile range, and cook's distance and performed outliers treatment by capping the outliers. And also removed some outliers

6. Perform certain variable selection methods such as factorial analysis also called an exploratory factorial analysis. On a covariance matrix or data frame, the function performs maximum-likelihood factor analysis. The claim factors specify the number of factors to be fitted to the machine learning algorithm. As shown in Fig 12

7. Finally, the dataset split into test, train, and fitted to certain machine learning algorithms and the outcome is evaluated using a specific evaluation technique

4.1 Dataset-1 – Video Game copy sales and rating

4.1.1 Data Description

This dataset is obtained from Kaggle. It consists of 16 columns and 16,720 rows. This data is web scraped by Gregory Smith from the website of VGChartz Video Games Sales. Some of the variables are web scraped from Metacritic and merged to prepare the final dataset.

Data collection source: Video Games Sales Dataset | Kaggle

4.1.2 Data Overview:

The dataset has 16 columns and 16,720 rows

4.1.3 Target Variable: From this dataset 2 research questions are formed. One is predicting the global sales percentage and the other one is predicting the rating of the games.

Dependent variable 1: Global_sales

Dependent variable 2: Rating

4.1.4 Null Values:

All the missing value in the data is visualized using the heatmap. After identifying the missing values imputation is done. Figure 2 illustrates the number of missing values in the data. This dataset has 13.2% of missing values and it needs to be imputed or dropped

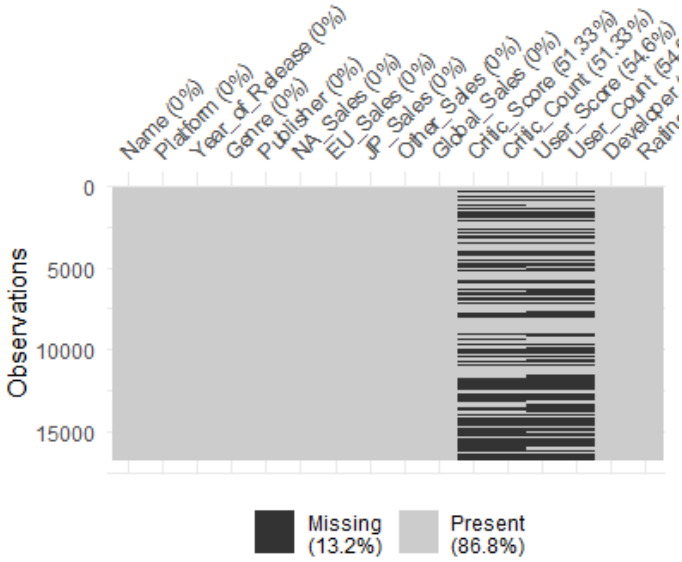

Figure 2: heatmap of missing values in dataset 1

4.1.5 Data Exploration & Cleaning:

After identifying the missing values and what are all the columns that hold a maximum number of missing values. Imputation is done and the outliers are checked. All the categorical variables are changed as factors and the levels are checked.

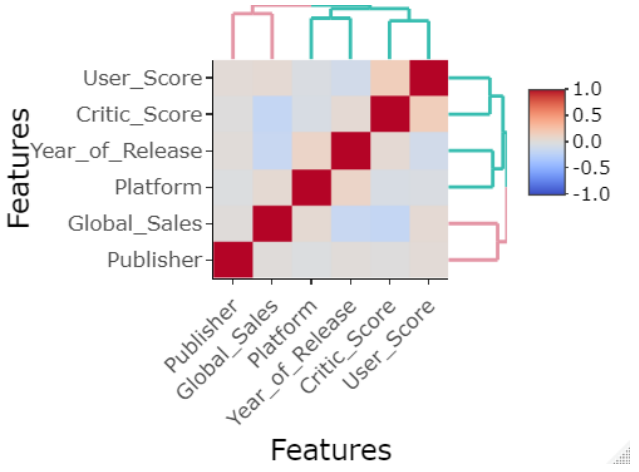

Figure 3: heatmap of missing values in dataset 3

4.1.6 Missing Numerical Values (Null values):

Method: Mice imputation

All the numerical variables are loaded to a separate data frame and imputation is done. Mice imputation algorithm is used to impute all the continuous variables. First, the correlation of the continuous variable that doesn't have a missing value is checked since the mice algorithm imputes the missing values using linear regression. The correlation matric of the data frame is shown in figure 3. After checking the correlation between the continuous variables all the continuous variables which have the missing values are loaded to the same data frame where the continuous variables which don't have null value exist and the imputation is performed. Since the linearity is not high some variables are imputed with a missing forest algorithm based on a random forest algorithm.

4.1.7 Missing Categorical Values (Null values):

Method: Missing forest

All the missing values are imputed using the missing forest algorithm and since the data matrix doesn't have much correlation some continuous variables are also imputed using random forest. It predicts missing values using a random forest trained on the observed values of a data matrix. Finally, all the null values are imputed in the data matrix. As shown in Fig 4

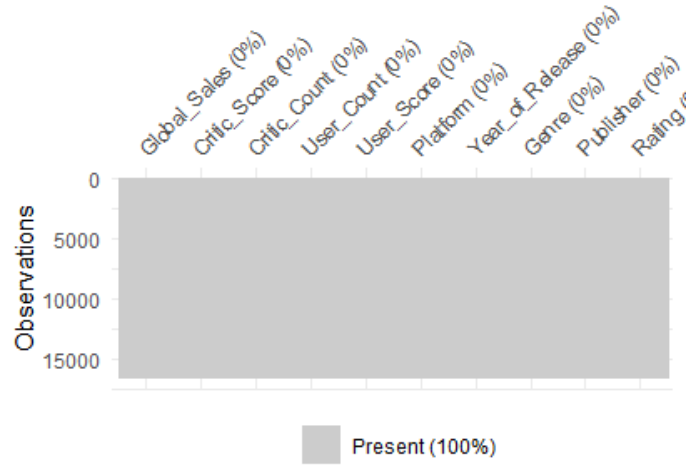

Figure 4: heatmap after imputation process

4.1.8 Outliers for Numerical Variables:

After all, the null values are imputed in the data frame matrix. Outliers in all the numerical values are checked by plotting boxplot and visualizing all the variables individually. Figure 5 explains the outliers in the numerical variables of dataset 1.

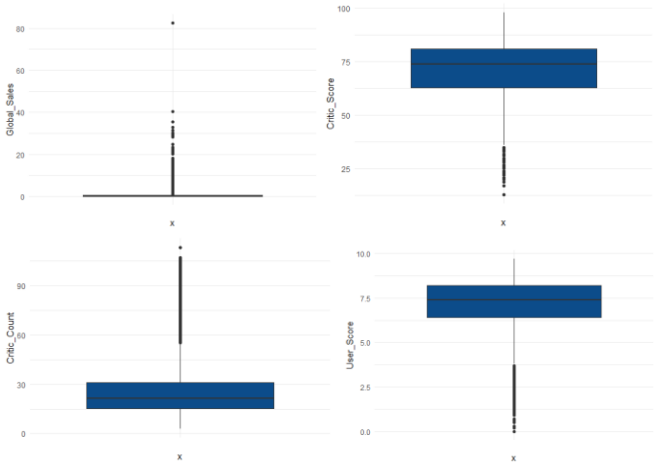

Figure 5: Outliers in dataset 1

4.1.9 Removing Outliers:

Method: Outlier capping

To remove the outliers all the outliers outlier capping method is used to impute all the outliers with quantiles. In this dataset, the value above the quartile is capped. After all the outliers are imputed using the outlier capping methods the remaining small amount of outliers are removed and one column is dropped to reduce the loss of data by dropping all the outliers in all the columns. Figure 6 shows the box plot of all the numerical variables after capping the outliers and dropping some remaining outliers

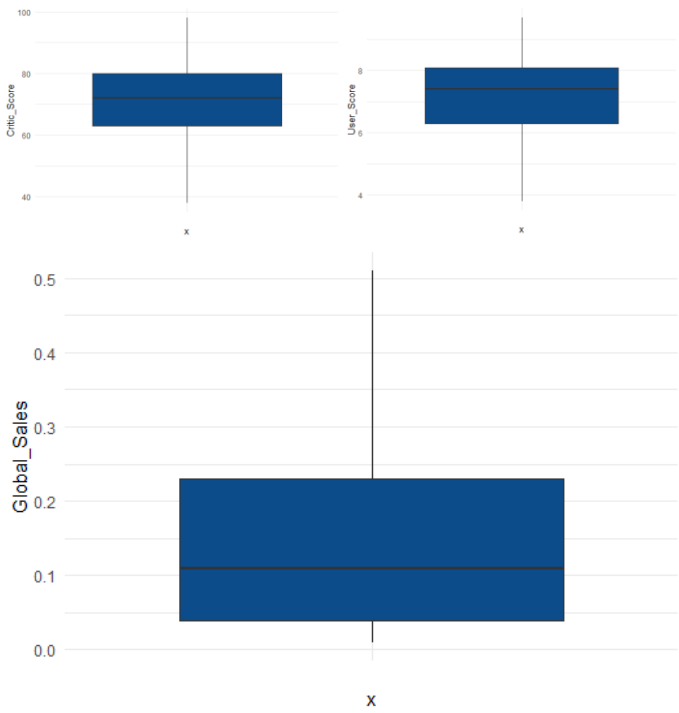

Figure 6: boxplot after capping the outliers

4.1.10 Encoding Categorical Variables:

Some categorical variable has more than 15 levels so it been converted to numerical variables by creating dummy variables such as 0, 1, 2 …, for all the categorical levels

4.1.11 Model Selection:

Since it has two research questions one dependent variable is continuous so the Random Forest regression model is built. And another dependent variable is the categorical variable so Random forest, KNN, and SVM algorithm is built and all the model performance is compared using different evaluation methods.

4.2 Dataset-2 – Apple app store games review and rating

4.2.1 Data Description:

This dataset is obtained from Kaggle. It consists of 16 columns and 17008 rows. On the Apple App Store, there are 17007 strategy games. It was gathered on August 3rd, 2019, with the aid of the iTunes API and the App Store sitemap.

Data collection source:
https://www.kaggle.com/tristan581/17k-apple-app-store-strategy-games

4.2.2 Data Overview:

The dataset has 17 columns and 17,007 rows

4.2.3 Target Variable:

From this dataset 1 research question is formed. One is predicting the global sales percentage and the other one is predicting the Average User Rating

.

Dependent variable 1: Average User Rating

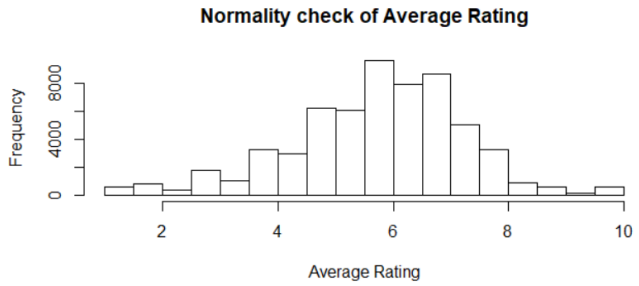

Figure 7: Normality of dependent variable

4.2.4 Null Values:

First, the empty cells in the data frame are filled with the N/A value. This is done because the missing values should be recognized by the function of the program. After filling all the empty cells with the N/A value the number of missing values is identified using the heatmap. The heatmap is shown in figure 8 explains the percentage of missing values in the dataset before preprocessing.

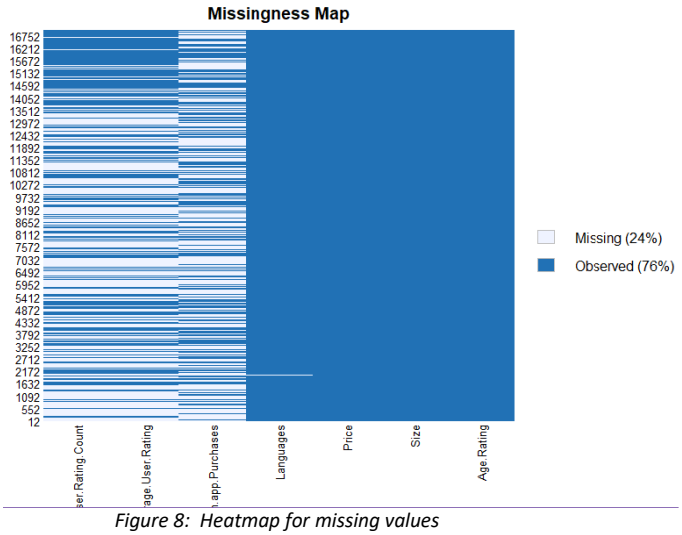

Figure 8: Heatmap for missing values

4.2.5 Data Exploration & Cleaning:

All the unwanted scrap variables are dropped from the data frame and load to a new data frame to create a clean resultant dataset to fit the machine learning algorithm. The correlation is shown in Fig 9.

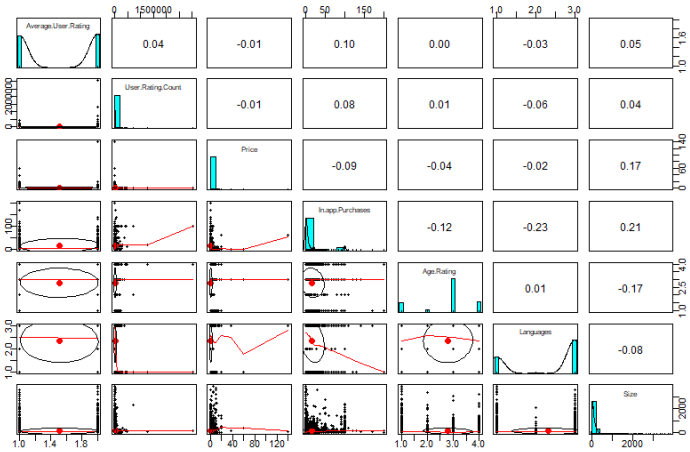

Figure 9: Correlation and distribution of variables in dataset 2

4.2.6 Missing Values (Null values):

The null values are dropped from the dataset in each column and the resultant dataset is shown below using the heatmap. Some of the missing values in the app purchase column are turned to 0 since 0 is the frequently occurring value.

```
# Converting all NA in In.App.Purchases to 0
df_3.df$In.app.Purchases[is.na(df_3.df$In.app.Purchases)] <- "0"
```

4.2.7 Outliers for Numerical Variables:

Influential data points that may negatively affect your regression model are identified using the cook's distance. The results of the cook's distance show that there are no outliers in the data.

$$D_i = (r_i^2 / p*MSE) * (h_{ii} / (1-h_{ii})^2)$$

Figure 10: formula for cook's distance

4.2.8 Encoding Variables :

The dependent categorical variable, **the** average rating is categorized into two values as good and bad. This is done to fit the dataset to the logistic regression classification model

```
blr_test_predict[blr_test_predict<=0.5] <- "Bad"
blr_test_predict[blr_test_predict != "Bad"] <- "Good"
blr_test_predict <- factor(blr_test_predict,
levels=c("Bad","Good"))
```

The column language is transformed into three levels to fit the machine learning algorithm. Reducing levels helps to optimize the performance of the classifier machine learning algorithm. To achieve optimization this reduction of levels are performed

```
# converting languages to have only 3 levels
conlang <- function(x){
 if (str_detect(x, "^.*EN.*$", negate = TRUE)) {
  varlan = "No EN"
 }
 else if (x == "EN"){
  varlan = "Only EN"
 }
 else {
  varlan = "EN +"
 }
 return(varlan)
}
```

In column ‘size’ the value is converted from KB to MB for easy understanding

```
# converting column size to MB from Bytes for easier
understanding
btomb <- function(x){
 varnum <- x/1048576
 varnum <- as.numeric(format(round(varnum, 2), nsmall = 2))
 return (varnum)
}
```

4.2.9 Model Selection:

This model is fitted to the logistic regression model to predict whether the rating is ‘Good’ or ‘Bad’.

4.3 Dataset-3 – Stream store game price and ratings

4.3.1 Data Description:

This dataset is obtained from Kaggle. It consists of 18 columns and 2707 rows. This dataset contains details about different aspects of games on the store, such as genre and an approximate number of owners, and is derived from the Steam Store and SteamSpy APIs.

4.3.2 Data Overview:

The dataset has 18columns and 27076rows

4.3.3 Target Variable:

Forecasting the game's success and the number of owners by average ratings of these games

Dependent variable 1: Number of owners

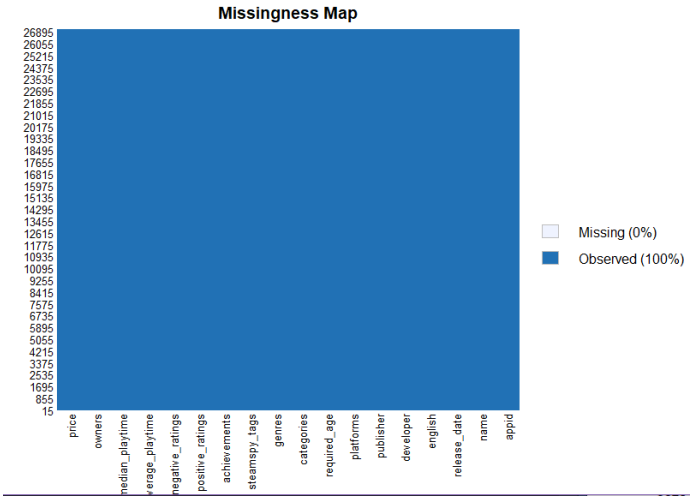

Figure 11: Heatmap of Data 3

4.3.4 Null Values:

This dataset has no missing values. It can be seen in the heatmap of figure 11.

4.3.5 Data Exploration & Cleaning:

Since it had no null values there is no need to impute using any other algorithm. All the unwanted scrap variables are dropped from the data frame and loaded to the same data frame.

4.3.6 Outliers for Numerical Variables:

Investigating all the numerical variable for outliers using cook’s distance and concluded that there are no outliers in the dataset

4.3.7 Encoding Variables :

Some variables are transformed into the factors and the levels are checked. Some of the factor levels are renamed and the levels are reduced to achieve the optimal performance after fitting the machine learning algorithm.

```
# renaming factor levels
levels(dmml.1.df$english)[levels(dmml.1.df$english) == 0] <-
"No"
levels(dmml.1.df$english)[levels(dmml.1.df$english) == 1] <-
"Yes"
levels(dmml.1.df$required_age)[levels(dmml.1.df$required_ag
e) == 0] <- "No Age Limit"
levels(dmml.1.df$required_age)[levels(dmml.1.df$required_ag
e) == 3] <- "3+"
levels(dmml.1.df$required_age)[levels(dmml.1.df$required_ag
e) == 7] <- "7+"
levels(dmml.1.df$required_age)[levels(dmml.1.df$required_ag
e) == 12] <- "12+"
levels(dmml.1.df$required_age)[levels(dmml.1.df$required_ag
e) == 16] <- "16+"
levels(dmml.1.df$required_age)[levels(dmml.1.df$required_ag
e) == 18] <- "18+"
levels(dmml.1.df$owners)[levels(dmml.1.df$owners) %in%
c("0-20000")] <- "<20K"
```

```
levels(dmml.1.df$owners)[levels(dmml.1.df$owners) %in% c("20000-50000","50000-100000","100000-200000","200000-500000")] <- "20K to 500K"
levels(dmml.1.df$owners)[levels(dmml.1.df$owners) %in% c("500000-1000000","1000000-2000000","2000000-5000000","5000000-10000000")] <- "500K to 10M"
levels(dmml.1.df$owners)[levels(dmml.1.df$owners) %in% c("10000000-20000000","20000000-50000000","50000000-100000000","100000000-200000000")] <- "10M to 200M"
levels(dmml.1.df$platforms)[levels(dmml.1.df$platforms) %in% c("mac;linux")] <- "mac linux"
levels(dmml.1.df$platforms)[levels(dmml.1.df$platforms) %in% c("windows;linux")] <- "windows linux"
levels(dmml.1.df$platforms)[levels(dmml.1.df$platforms) %in% c("windows;mac")] <- "windows mac"
levels(dmml.1.df$platforms)[levels(dmml.1.df$platforms) %in% c("windows;mac;linux")] <- "windows mac linux"
```

After transforming certain column categories by extracting certain data and resetting the row count is performed.

4.3.8 Model Selection:

This model is fitted to two machine algorithm and the performance of the two variables are compared. The machine learning model used are Decision tree and RIPPER RULE algorithm also called Repeated Incremental Pruning to Produce Error Reduction

V.1 MODEL FITTING AND EVALUATION

5.1 Dataset-1 – Video Game copy sales and rating

```
factanal(x = numeric_data, factors = 3)

Uniquenesses:
       Platform Year_of_Release       Publisher    Global_Sales    Critic_Score      User_Score
          0.969           0.005           0.995           0.642           0.827           0.521

Loadings:
              Factor1 Factor2 Factor3
Platform        0.140           0.106
Year_of_Release 0.976          -0.205
Publisher
Global_Sales            0.153   0.577
Critic_Score            0.210  -0.358
User_Score     -0.126   0.674

               Factor1 Factor2 Factor3
SS loadings      0.991   0.525   0.524
Proportion Var   0.165   0.088   0.087
Cumulative Var   0.165   0.253   0.340

The degrees of freedom for the model is 0 and the fit was 5e-04
```

Figure 12: Exploratory factorial analysis of Dataset 1

The factorial analysis is done to select the variables

5.1.1 MODEL 1: Random forest regression model

Prediction And Analysis: The Random forest regression model has the highest precision of (33%) with an RMSE(root mean square deviation) of 0.1 shown in Fig 13

Evaluation:

```
#Model performance metrics
##########################

data.frame(R2 = R2(predict_1, test$Global_Sales),
           RMSE = RMSE(predict_1, test$Global_Sales),
           MAE = MAE(predict_1,test$Global_Sales))
        R2      RMSE       MAE
0.3307552 0.1068365 0.08050801
```

Figure 13: Evaluation of random forest regression model

5.1.2 MODEL 2: Random forest classifier model

Prediction And Analysis:

The Random forest Classifier model has the highest precision of (63%)
Test accuracy: 63%
Confusion Matrix: Figure 14
Classification Report: Figure 14
ROC-AUC curve: 15

Evaluation:

```
Confusion Matrix and Statistics

          Reference
Prediction    E E10+   EC  K-A    M   RP    T
      E    1101  196    1    0   43    0  260
      E10+   57  159    0    0   15    0   34
      EC      0    0    0    0    0    0    0
      K-A     0    0    0    0    0    0    0
      M      15   12    0    0  117    0   55
      RP      0    0    0    0    0    0    0
      T     182  123    1    0  162    1  676

Overall Statistics

               Accuracy : 0.6396
                 95% CI : (0.6227, 0.6562)
    No Information Rate : 0.4221
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4509

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: E Class: E10+ Class: EC Class: K-A Class: M Class: RP Class: T
Sensitivity            0.8125     0.32449 0.0000000         NA  0.34718 0.0000000   0.6595
Specificity            0.7305     0.96103 1.0000000          1  0.97146 1.0000000   0.7854
Pos Pred Value         0.6877     0.60000       NaN         NA  0.58794       NaN   0.5904
Neg Pred Value         0.8421     0.88761 0.9993769         NA  0.92693 0.9996885   0.8310
Prevalence             0.4221     0.15265 0.0006231          0  0.10498 0.0003115   0.3193
Detection Rate         0.3430     0.04953 0.0000000          0  0.03645 0.0000000   0.2106
Detection Prevalence   0.4988     0.08255 0.0000000          0  0.06199 0.0000000   0.3567
Balanced Accuracy      0.7715     0.64276 0.5000000         NA  0.65932 0.5000000   0.7224
```

Figure 14: Evaluation of random forest classification model

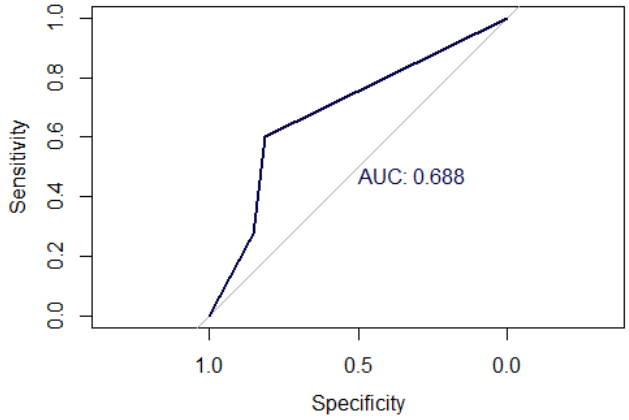

Figure 15: ROC-AUC curve random forest classification model

5.1.3 MODEL 3: SVM(*Support Vector Machine*) classifier model

Prediction And Analysis:

The SVM Classifier model has the highest precision of (53%)
Test accuracy: 53%
Confusion Matrix: Figure 16
Classification Report: Figure 16
ROC-AUC curve: Figure 17

Evaluation:

```
Confusion Matrix and Statistics

          Reference
Prediction    E E10+   EC  K-A    M   RP    T
      E    1187  325    0    1  106    0  427
      E10+   44   46    0    0    7    0   52
      EC      0    0    0    0    0    0    0
      K-A     0    0    0    0    0    0    0
      M      16   16    0    0   32    0   30
      RP      0    0    0    0    0    0    0
      T     372  196    1    0  243    3  749

Overall Statistics

               Accuracy : 0.5227
                 95% CI : (0.5068, 0.5386)
    No Information Rate : 0.4202
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.2496

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: E Class: E10+ Class: EC Class: K-A Class: M Class: RP Class: T
Sensitivity            0.7332     0.07890 0.0000000  0.0000000 0.082474 0.0000000  0.5954
Specificity            0.6155     0.96850 1.0000000  1.0000000 0.982107 1.0000000  0.6859
Pos Pred Value         0.5802     0.30872       NaN        NaN 0.340426        NaN  0.4789
Neg Pred Value         0.7609     0.85502 0.9997405  0.9997405 0.905294 0.9992214  0.7776
Prevalence             0.4202     0.15131 0.0002595  0.0002595 0.100701 0.0007786  0.3265
Detection Rate         0.3081     0.01194 0.0000000  0.0000000 0.008305 0.0000000  0.1944
Detection Prevalence   0.5310     0.03867 0.0000000  0.0000000 0.024397 0.0000000  0.4059
Balanced Accuracy      0.6743     0.52370 0.5000000  0.5000000 0.532291 0.5000000  0.6407
```

Figure 16: Evaluation of SVM classification model

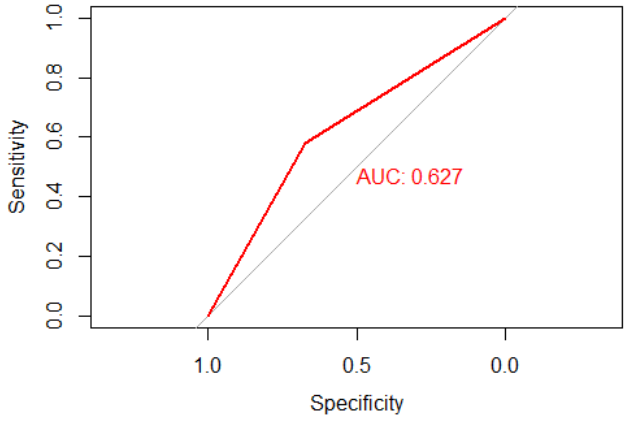

Figure 17: ROC-AUC curve of SVM classification model

5.1.4 MODEL 4: KNN(*K- Nearest Neighbours*) classifier model

Prediction And Analysis:

The *K- Nearest Neighbours* Classifier model has the highest precision of (45%)
Test accuracy: 45%
Confusion Matrix: Figure 18
Classification Report: Figure 18
ROC-AUC curve: Figure 19

Evaluation:

```
Confusion Matrix and Statistics

          Reference
Prediction   E E10+  EC   M  RP   T
      E    915  206   1 134   1 523
      E10+   0    0   0   0   0   0
      EC     0    0   0   0   0   0
      M      0    0   0   0   0   0
      RP     0    0   0   0   0   0
      T    440  284   1 203   0 502

Overall Statistics

               Accuracy : 0.4414
                 95% CI : (0.4242, 0.4588)
    No Information Rate : 0.4221
    P-Value [Acc > NIR] : 0.0141

                  Kappa : 0.1044

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: E Class: E10+ Class: EC Class: M Class: RP Class: T
Sensitivity            0.6753       0.0000 0.0000000     0.000 0.0000000  0.4898
Specificity            0.5337       1.0000 1.0000000     1.000 1.0000000  0.5753
Pos Pred Value         0.5140          NaN        NaN       NaN        NaN  0.3510
Neg Pred Value         0.6923       0.8474 0.9993769     0.895 0.9996885  0.7062
Prevalence             0.4221       0.1526 0.0006231     0.105 0.0003115  0.3193
Detection Rate         0.2850       0.0000 0.0000000     0.000 0.0000000  0.1564
Detection Prevalence   0.5545       0.0000 0.0000000     0.000 0.0000000  0.4455
Balanced Accuracy      0.6045       0.5000 0.5000000     0.500 0.5000000  0.5325
```

Figure 18: Evaluation of KNN classification model

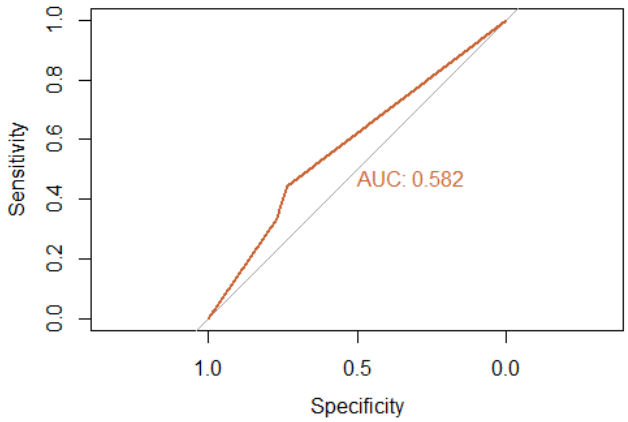

Figure 19: ROC-AUC curve KNN classification model

5.2Dataset-2 – Apple app store games review and rating

5.2.1 MODEL 1: Logistic regression model

Prediction And Analysis:

The Logistic regression Classifier model has the highest precision of (57%)
Test accuracy: 57%
Confusion Matrix: Figure 20
Classification Report: Figure 20
ROC-AUC curve: Figure 21

Evaluation:

```
Confusion Matrix and Statistics

          Reference
Prediction Bad Good
      Bad  814  657
      Good 304  486

               Accuracy : 0.575
                 95% CI : (0.5543, 0.5955)
    No Information Rate : 0.5055
    P-Value [Acc > NIR] : 2.103e-11

                  Kappa : 0.1528

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.4252
            Specificity : 0.7281
         Pos Pred Value : 0.6152
         Neg Pred Value : 0.5534
             Prevalence : 0.5055
         Detection Rate : 0.2149
   Detection Prevalence : 0.3494
      Balanced Accuracy : 0.5766

       'Positive' Class : Good
```

Figure 20: Evaluation of logistic regression classification model

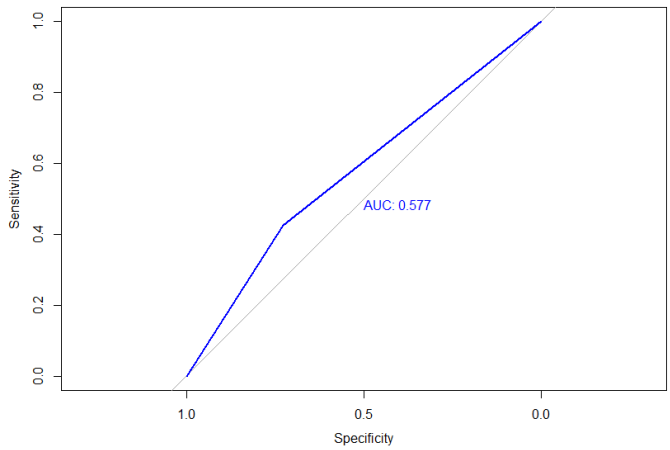

Figure 21: ROC-AUC curve logistic regression classification model

5.3 Dataset-3 – Stream store game price and ratings

5.3.1 MODEL 1: Decision tree classification model

Prediction And Analysis:

The Decision tree Classifier model has the highest precision of (89%)
Test accuracy: 89%
Confusion Matrix: Figure 22
Classification Report: Figure 22
ROC-AUC curve: Figure 24

Evaluation:

```
Confusion Matrix and Statistics

             Reference
Prediction    <20K 20K to 500K 500K to 10M 10M to 200M
  <20K        4386         359           1           0
  20K to 500K  244        1455          69           0
  500K to 10M    0          53         195           4
  10M to 200M    0           0           0           3

Overall Statistics

               Accuracy : 0.8922
                 95% CI : (0.8845, 0.8994)
    No Information Rate : 0.684
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7587

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: <20K Class: 20K to 500K Class: 500K to 10M Class: 10M to 200M
Sensitivity               0.9473              0.7793             0.73585           0.4285714
Specificity               0.8317              0.9361             0.99124           1.0000000
Pos Pred Value            0.9241              0.8230             0.77381           1.0000000
Neg Pred Value            0.8794              0.9176             0.98926           0.9994088
Prevalence                0.6840              0.2758             0.03915           0.0010341
Detection Rate            0.6480              0.2150             0.02881           0.0004432
Detection Prevalence      0.7011              0.2612             0.03723           0.0004432
Balanced Accuracy         0.8895              0.8577             0.86354           0.7142857
```

Figure 22: Decision Tree classification model

5.3.2 MODEL 2: RIPPER RULE classification model

Prediction And Analysis:

The RIPPER RULE model has the highest precision of (89.3%)
Test accuracy: 89.3%
Confusion Matrix: Figure 23
Classification Report: Figure 23

ROC-AUC curve: Figure 24

Evaluation:

```
Confusion Matrix and Statistics

             Reference
Prediction    <20K 20K to 500K 500K to 10M 10M to 200M
  <20K        4424         386           1           0
  20K to 500K  206        1420          60           0
  500K to 10M    0          61         201           2
  10M to 200M    0           0           3           5

Overall Statistics

               Accuracy : 0.8938
                 95% CI : (0.8862, 0.901)
    No Information Rate : 0.684
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7606

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: <20K Class: 20K to 500K Class: 500K to 10M Class: 10M to 200M
Sensitivity               0.9555              0.7606             0.75849           0.7142857
Specificity               0.8191              0.9457             0.99031           0.9995563
Pos Pred Value            0.9196              0.8422             0.76136           0.6250000
Neg Pred Value            0.8948              0.9121             0.99016           0.9997042
Prevalence                0.6840              0.2758             0.03915           0.0010341
Detection Rate            0.6536              0.2098             0.02969           0.0007387
Detection Prevalence      0.7107              0.2491             0.03900           0.0011819
Balanced Accuracy         0.8873              0.8532             0.87440           0.8569210
```

Figure 23: Ripper Rule classification model

COMPARING MODEL 1 AND MODEL 2 OF DATA 3

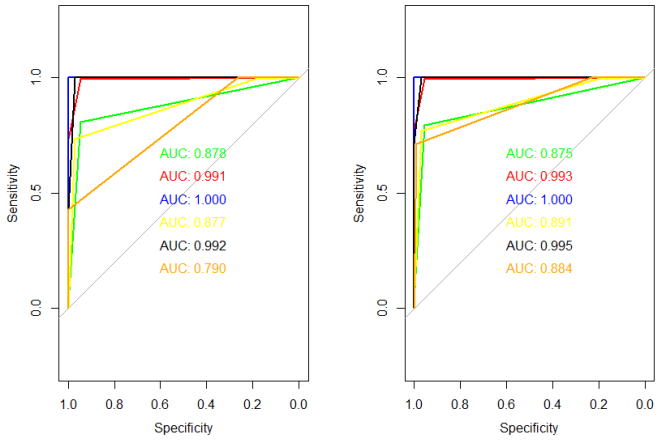

Figure 24: ROC-AUC curve Ripper rule and decision tree classification model

VI. CONCLUSION AND FUTURE WORK

Data was imported, filtered, analyzed, explored, and transformed before being fed into various models that assisted in answering various questions. When compared to KNN and SVM, the Random forest Classifier approach proved to be a better model for predicting game ratings in Dataset-1, with a model accuracy of 63 percent. For Dataset-2, the logistic regression model had a model accuracy of 58 percent and had the best results. Both models in Dataset 3 such as the Decision tree and the Ripper rule models were best, with an overall precision of 89.22 percent and 89.32 percent, respectively. The models for Dataset-3 cannot be compared since they both achieved the best performance.

For future work, the best dimensional reduction algorithm must be used to select the best subset from the data to fit into the machine learning algorithms. Tuning of algorithms concerning the dataset should be done to achieve maximum accuracy. Using different evaluation techniques and normalization of data.

REFERENCES

[1] S. Ghosh and C. Banerjee, "A Predictive Analysis Model of Customer Purchase Behavior using Modified Random Forest Algorithm in Cloud Environment," 2020 IEEE 1st International Conference for Convergence in Engineering (ICCE), 2020, pp. 239-244, doi: 10.1109/ICCE50343.2020.9290700.

[2] P. Leng et al., "Logistic Regression Based on Artificial Fish Swarm Algorithm with T-Distribution Parameters," 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), 2020, pp. 1912-1915, doi: 10.1109/ITAIC49862.2020.9338804.

[3] S. Dhali, M. Pati, S. Ghosh and C. Banerjee, "An Efficient Predictive Analysis Model of Customer Purchase Behavior using Random Forest and XGBoost Algorithm," 2020 IEEE 1st International Conference for Convergence in Engineering (ICCE), 2020, pp. 416-421, doi: 10.1109/ICCE50343.2020.9290576.

[4] H. He et al., "Ensemble MultiBoost Based on RIPPER Classifier for Prediction of Imbalanced Software Defect Data," in IEEE Access, vol. 7, pp. 110333-110343, 2019, doi: 10.1109/ACCESS.2019.2934128.

[5]H. Xu, "Prediction on Bundesliga Games Based on Decision Tree Algorithm," 2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), 2021, pp. 234-238, doi: 10.1109/ICBAIE52039.2021.9389986.

[6]D. Draskovic, M. Brzakovic and B. Nikolic, "A comparison of machine leaming methods using a two player board game," IEEE EUROCON 2019 -18th International Conference on Smart Technologies, 2019, pp. 1-5, doi: 10.1109/EUROCON.2019.8861927.

[7] Z. Liu, D. Ma and Z. Feng, "A Feature Selection Algorithm Based on SVM Average Distance," 2010 International Conference on Measuring Technology and Mechatronics Automation, 2010, pp. 90-93, doi: 10.1109/ICMTMA.2010.792.

[8] R. Serban, A. Kupraszewicz and G. Hu, "Predicting the characteristics of people living in the South USA using logistic regression and decision tree," 2011 9th IEEE International Conference on Industrial Informatics, 2011, pp. 688-693, doi: 10.1109/INDIN.2011.6034974.

[9] A. I. Rathnayake, I. Kumari Ganegala, I. S. Samarasinghe, S. Bandara Weerasekara, A. I. Gamage and T. Thilakarathna, "Adjusting The Hard Level on Game by a Prediction for Improving Attraction and Business Value," 2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer), 2020, pp. 310-311, doi: 10.1109/ICTer51097.2020.9325474.

[10] C.M. Wu, P. Patil and S. Gunaseelan, "Comparison of Different Machine Learning Algorithms for Multiple Regression on Black Friday Sales Data," 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), 2018, pp. 16-20, doi: 10.1109/ICSESS.2018.8663760.

[11] S. Kaya and M. Yağanoğlu, "An Example of Performance Comparison of Supervised Machine Learning Algorithms Before and After PCA and LDA Application: Breast Cancer Detection," 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), 2020, pp. 1-6, doi: 10.1109/ASYU50717.2020.9259883.

[12] A. El-Koka, K. Cha and D. Kang, "Regularization parameter tuning optimization approach in logistic regression," 2013 15th International Conference on Advanced Communications Technology (ICACT), 2013, pp. 13-18.

[13] Cui, X., Wang, B., Wang, L., Chen, J. **Online optimal learning algorithm for Stackelberg games with partially unknown dynamics and constrained inputs** (2021) *Neurocomputing*, 445, pp. 1-11.

[14] Özer, Ç., Çevik, T. & Gürhanlı, A. A machine learning-based framework for predicting game server load. *Multimed Tools Appl* **80,** 9527–9546 (2021). https://doi.org/10.1007/s11042-020-10067-5

[15] Liu, H., Zhang, Y., **Bridge condition rating data modeling using deep learning algorithm** (2020) *Structure and Infrastructure Engineering*, 16 (10), pp. 1447-1460. Cited4times.
DOI: 10.1080/15732479.2020.1712610, Department of Civil & Environmental Engineering, University of Maryland, College Park, MD, United States

[16] P. Nair and I. Kashyap, "Hybrid Pre-processing Technique for Handling Imbalanced Data and Detecting Outliers for KNN Classifier," 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), 2019, pp. 460-464, doi: 10.1109/COMITCon.2019.8862250.

[17] G. Wu and J. Tao, "Design and application of machine learning algorithm computer in connect6 of computer games system," 2016 Chinese Control and Decision Conference (CCDC), 2016, pp. 4279-4282, doi: 10.1109/CCDC.2016.7531734.

[18] P. Leng et al., "Logistic Regression Based on Artificial Fish Swarm Algorithm with T-Distribution Parameters," 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), 2020, pp. 1912-1915, doi: 10.1109/ITAIC49862.2020.9338804.

[19] J. Lee and T. Yoon, "Analysis of relation between aging and telomere using datamining — Apriori, decision tree, and Support Vector Machine(SVM)," 2017 19th International Conference on Advanced Communication Technology (ICACT), 2017, pp. 685-689, doi: 10.23919/ICACT.2017.7890180.

[20] Peng Li, Siben Li, Tingting Bi and Yang Liu, "Telecom customer churn prediction method based on cluster stratified sampling logistic regression," International Conference on Software Intelligence Technologies and Applications & International Conference on Frontiers of Internet of Things 2014, 2014, pp. 282-287, doi: 10.1049/cp.2014.1576.