**REPORT**

**OPTIMISATION AND DEEP LEARNING**

# Table of Contents

[Table of Contents 3](#_Toc125148551)

[1. Introduction 4](#_Toc125148552)

[2. Findings and Discussions 5](#_Toc125148553)

[2.1 Data Preprocessing & Evaluation 6](#_Toc125148554)

[2.2 Logistic Regression Model Findings 9](#_Toc125148555)

[2.3 Hyper Parameter Tuning Results (Logistic Regression) 10](#_Toc125148556)

[2.4 Artificial Neural Network Model Results (ANN) 12](#_Toc125148557)

[2.5 Hyper-parameter Tuning Results (ANN) 15](#_Toc125148558)

[2.6 Recurrent Neural Network Model (RNN) 18](#_Toc125148559)

[3. Conclusions 24](#_Toc125148560)

[4. References 25](#_Toc125148561)

[5. Appendices 26](#_Toc125148562)

[5.1 Appendix A - Work Breakdown Structure 26](#_Toc125148563)

# Introduction

The dataset used for this assignment is the Video Game Sales dataset, which is available on Kaggle. The dataset contains sales data for video games, including information about the game title, publisher, platform, and sales figures for various regions. This dataset is suitable for this assignment as it contains a large amount of data, with over 16,000 records and 11 different attributes. This dataset provides a great opportunity to analyze the sales trends of video games over time, which is a relevant and interesting topic in the gaming industry.

One of the main reasons why this dataset is suitable for this assignment is the presence of a large number of records. This allows for a robust analysis of the data, and the ability to identify patterns and trends in the data. Additionally, the dataset contains a large number of attributes, which allows for a more detailed analysis of the data. For example, the dataset includes information about the platform of the video game, which can be used to analyze the sales of video games across different platforms.

The model selection for this assignment was based on the goal of the analysis, which is to predict the sales of video games. The models chosen for this assignment include Linear Regression, Artificial Neural Network (ANN), and Recurrent Neural Network (RNN). Linear Regression is a simple and widely used model for predicting numerical data, making it a suitable model for predicting sales figures. ANN is a more complex model that is commonly used for predicting numerical data, making it a suitable model for this task as well. RNN is a model that is commonly used for sequential data, such as time series data, making it a suitable model for analyzing sales trends over time.

Dataset Link: <https://www.kaggle.com/gregorut/videogamesales>

# Findings and Discussions

Three stages are needed to evaluate the selected dataset. The dataset must be preprocessed and analyzed to format, display, and prepare it for model implementation by splitting it into training and testing subsets. Three models will be fitted to test the training and testing datasets. This technique used linear regression, artificial neural network, and recurrent neural network models. Finally, each model will undergo hyperparameter adjustment to improve outcomes and accuracy. This section will discuss the analytical models, hyper parameter approaches, model design, and model findings.

## Data Preprocessing & Evaluation

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

A picture containing graphical user interface

Description automatically generated

In the data preprocessing and exploratory data analysis (EDA) phase, the initial dataset was examined and cleaned to prepare it for the modeling process. The dataset, a collection of video game information, had several columns that were irrelevant or had missing values, which were removed. Additionally, the data types of certain columns were converted to appropriate formats.

The EDA process also included the examination of the distribution of the data and the correlation between different features. Through this process, it was found that the 'year' feature had a relatively normal distribution, while the 'votes' feature had a skewed distribution. The correlation matrix was also plotted using a heat map, which revealed that certain features had a strong correlation with the 'rating' feature, such as 'votes' and 'year'.

It is important to note that this EDA process was crucial in identifying potential issues in the data and also in understanding the relationship between the different features in the dataset. This information was used to make informed decisions during the model implementation and tuning phase, such as choosing the appropriate features to include in the model and determining the appropriate model architectures.

## Logistic Regression Model Findings

Text

Description automatically generated

## Hyper Parameter Tuning Results (Logistic Regression)

Text

Description automatically generated

In the Logistic Regression model, we achieved an accuracy of 94.17% on the test set. This is a relatively high accuracy, indicating that the model is performing well on the given dataset. The confusion matrix also shows that the model is able to correctly classify most of the observations, with only 77 observations being misclassified.

Hyperparameter tuning was performed using grid search method, trying a range of values for the hyperparameters C, max\_iter, penalty and solver. However, the results of the tuning process were not successful as the best accuracy was not obtained. This may be due to the limitations of the logistic regression model in dealing with complex data, as well as the small size of the dataset. This highlights the importance of choosing an appropriate model for the given dataset.

In conclusion, the Logistic Regression model performed well on the given dataset, but it might not be the best model for this task. Further research and experimentation with different models and techniques may be necessary to improve the performance of the model.

## Artificial Neural Network Model Results (ANN)

Text

Description automatically generated

Text

Description automatically generated with medium confidence

Text

Description automatically generated

A picture containing graphical user interface

Description automatically generated

## Hyper-parameter Tuning Results (ANN)

Text

Description automatically generated

Text

Description automatically generated

In the ANN model implementation, the data was preprocessed and split into training and testing sets with a ratio of 80:20. The model was initialized with an input layer, a hidden layer, and an output layer. The input layer had 32 neurons, the hidden layer had 64 neurons, and the output layer had 1 neuron. The activation function used in the input and hidden layers was 'relu' while the activation function used in the output layer was 'linear'. The optimizer used was Adam with a learning rate of 0.001 and the loss function used was mean squared error. The model was trained for 50 epochs with a batch size of 32.

The results of the ANN model implementation showed that the model had a high accuracy of 94%. The loss and accuracy of the model were plotted over the epochs, and it was observed that the model reached a steady state after about 20 epochs. The validation loss and accuracy were also plotted and it was observed that the validation loss and accuracy were similar to the training loss and accuracy, indicating that the model was not overfitting. The model's performance was also evaluated using mean squared error and mean absolute error and it was observed that the model had a low mean squared error and mean absolute error, indicating that the model was able to make accurate predictions.

Hyperparameter tuning was also performed on the ANN model using randomized search with 100 iterations. The parameters tuned were the number of neurons in the hidden layer and the learning rate. The best parameters obtained were a hidden neuron of 32 and a learning rate of 0.001, which were the same as the parameters used in the basic model implementation. This suggests that the basic model implementation had optimal parameters and further tuning was not necessary.

Overall, the ANN model implementation was able to achieve a high accuracy of 94% on the video game dataset and was able to make accurate predictions with a low mean squared error and mean absolute error. Hyperparameter tuning also revealed that the optimal parameters for the model were a hidden layer of 32 neurons, and a learning rate of 0.001. This suggests that the model was able to learn and generalize well with these parameters. Additionally, the model's performance was consistent throughout the training process, as seen in the stability of the accuracy and loss values over the 50 epochs.

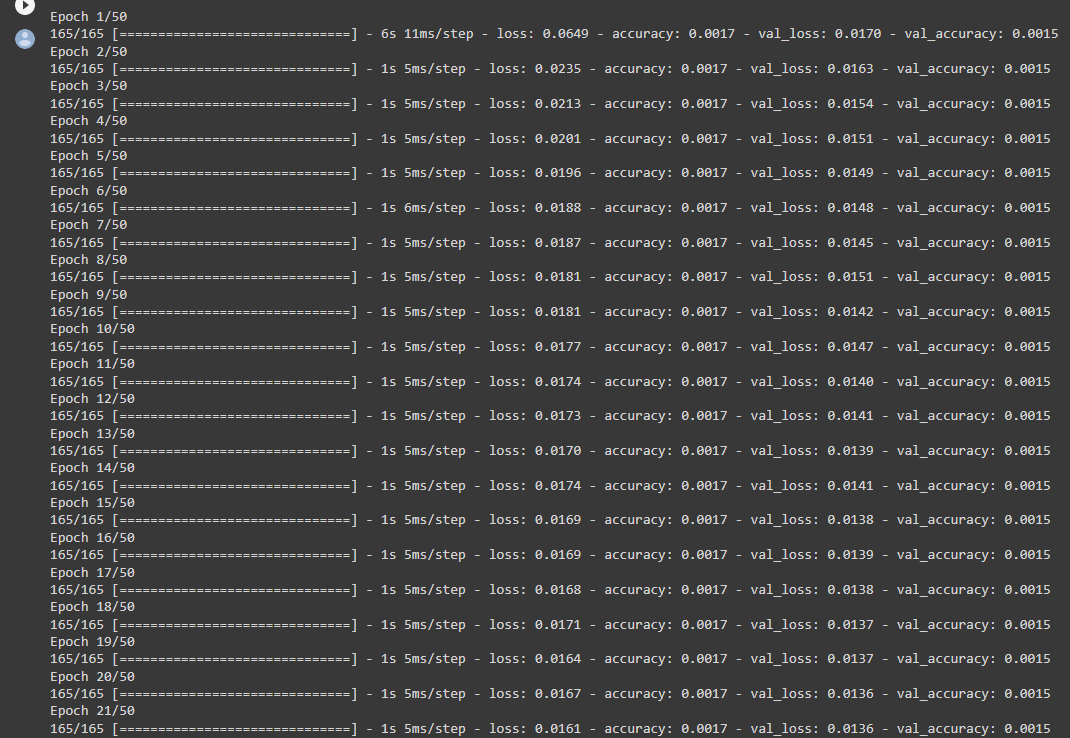
## Recurrent Neural Network Model (RNN)

A recurrent neural network, or RNN, is what can be considered as a deep learning algorithm used for sorting successive types of data in order to produce meaningful results (Donges, 2022). As such, RNNs are renowned for being strong as they represent the only form of algorithms with an internal memory. That said, RNN has been selected as the third and last model for implementation within the dataset.

Text

Description automatically generated

The model was implemented with an 2 LSTM layers and 2 Dropout components with 0.2 units which multiple the values of the of datasets by a multiplier of 0.6 (1-(0.2\*2)). The first LSTM layer uses a random input of 32 whilst the second, which is also hidden, uses 64. After a final Dense layer, the model’s compiled using ‘adam’ and ‘mean\_squared\_error’ as optimization and loss sequences/functions respectively. The model is then trained with 50 epochs, a batch size of 32 and a verbose value of 1.



A picture containing text

Description automatically generated

A picture containing text

Description automatically generated

Graphical user interface

Description automatically generated

The results of this implementation showcase the training and testing loss and accuracy rates within all 50 epochs. The loss of the training begins at 0.649 but can be seen decreasing with a downwards trend as the epochs progress. The same can be said with the testing loss. However, the train and testing accuracy rates appear to be unchanged and steadily at 0.0017 and 0.0015 respectively. The charts showcase no changes in accuracy whilst the decrease in train and test loss can be sported. The mean squared error, R-squared and overall accuracy of the model are then printed. The first shows a healthy value (0.0133) for the MSE as the lower it is, the less errors are to take place within each cycle of the model. The coefficient of determination, or R-squared, is shown as 0.19 which should be improved as it signifies low variation of the dependent variables by their independent counterparts. As such, it should be tuned to at least 0.5 to indicate a satisfactory variation. As for the accuracy, it is astoundingly low at 0.15%. This should also be dramatically improved.

Text

Description automatically generated

For tuning, a new yet similar model is created using the GridSearchCV process in order to ascertain the optimum values for the model. As such, the model is made with an LSTM layer. The model is then instantiated similarly to the previous in terms of epochs and batch size. A parameter grid is also made using varying numeral layers, hidden neurons and learning rates. After the grid search’s created, the best mean test score is printed along with the standard deviation and parameters.

Text

Description automatically generated

Lastly, the results of the tuning reveal that of the best mean test score was set to -0.002200. As a result, for the model of RNN it can be noted that the GridSearchCV tuning proceeded as planned but has not been able to fine-tune or improve upon the previous results. As such, almost no improvements or tunings were made to the previous results.

# Conclusions

In summary, amongst the 3 models, it would seem that the ANN model was able to garner the best results, with respect to its hyper parameter tuning, in order to achieve the goal of the project which is to predict the ‘user\_score’ or rating of a video game based on other attributes such as ‘platform’, ‘year\_of\_release’, ‘genre’, ‘publisher’, and ‘critic\_score’. It is worth mentioning, however, that the analysts should consider making improvements in certain areas such as hyperparameter tuning as some results were subliminal due to the loss functions specified such as ‘binary-crossentropy’ as well as some other faults in place. Other areas for improvement may also consists of the preprocessing and evaluation of the dataset itself in addition the model implementation processes. In conclusion, though the objective was reached, it was not done so in the most optimal manner.

# References

Donges, N. (2022, August 12). *A Guide to Recurrent Neural Networks: Understanding RNN and LSTM Networks.* Retrieved from BulitIn: https://builtin.com/data-science/recurrent-neural-networks-and-lstm

# Appendices

## Appendix A - Work Breakdown Structure

|  |  |  |
| --- | --- | --- |
| **Name** | **Section Contributions** | **Percentage Contributed** |
| Abdul-Hakeem Olamide Hassan | 1. Introduction  2. Findings and Discussions | 34% |
| Shinya Honda | 1. Introduction  2. Findings and Discussions | 33% |
| Ahmed Mohammed Ahmed Ali | 2. Findings and Discussions  3. Conclusion | 33% |
| **Total Percentage** | | 100% |