Machine learning and Data analysis

CW1 Report

Introduction

The following report will discuss the implementation of a supervised machine learning algorithm, providing a thorough summary of the processing of data as well as the implementation of the machine learning model, while also reporting on the accuracy of the implemented model. Information on the chosen dataset and steps required to sanitise this data to be used with a supervised machine learning model will be discussed while also going into detail the steps required to implement a support vector machine (SVM) model, a Neural network model or a deep learning model. Finally, the results of the chosen model will be reported, and any steps taken to tune or improve the results of the model will also be reported on with a short discussion on the importance of the data processing used in this report, as well as discussing the chosen machine learning model and what made it effective implemented with the chosen dataset.

Description of data and the problem

The chosen data structure contains information on in game statistics from the video game league of legends, consisting of information about how well players performed up until the 10-minute mark of a game. Each instance of the data structure counts as an individual game, with almost 10 thousands games in the dataset, containing almost 40 different columns all containing statistics on how well the two teams playing against each other in each game performed.

Graphical user interface, application, table, Excel

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

Player statistics by the 10-minute mark of each of these instances is collected, with the datasets variables containing:

* Kills
* First bloods
* Gold difference between teams
* Experience point difference between teams
* Deaths
* Assists
* Objectives
* Minions killed
* Etc

These are all very important pieces of information which all contribute overall to the outcome of a game of league of legends. These pieces of information can decide whether or not one team or the other team ends up winning a game of league of legends.

As outlined in figures 1 & 2, there are 9,879 instances with 40 variables in this dataset, entirely made up of numerical values without any null or empty values in the dataset.

The report problem is as follows: Can the chosen dataset be used to predict the outcome of a game of league of legends based on player statistics collected in the first 10 minutes of a game. Given the nature of the question and dataset, this can be considered a classification problem, meaning a support vector machine (SVM) algorithm would be suitable to solve the problem, which is a effective supervised machine learning algorithm. Supervised machine learning algorithms involve labelled data, with these algorithms being very good at predicting new data based on old data provided. (McGregor, 2020)

Data Pre-processing

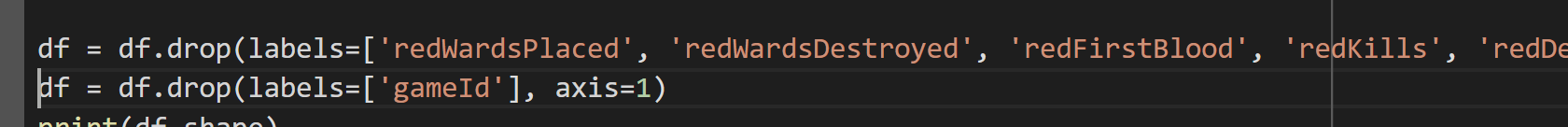
Before any model can be created to predict the outcome of a game based on data collected from the first 10 minutes, the data must be properly sanitised and processed, as this can ensure that the data isn’t volatile or unstructured in anyway that could negatively affect the SVM classifier.

Checking the dtypes of the dataset we can see that all data type are either an integer or a float, which is fine for an SVM model and does not need changed. Usually if there are any objects or strings present, they would need sanitised before it is used by the SVM classifier.

Text

Description automatically generated

Next any variables that are not needed for the classifier are removed, as data about the blueTeam and redTeam is available, however data about the redTeam is not necessary, as the aim of the model is to predict if data collected can predict a win or a loss for an individual team. This cleans up the dataset and reduces the amount of processing when it comes to constructing and tuning the SVM model.



Next the dataset must be checked for any null values and duplicate values, which there are none of in the dataset, so no processing is necessary at this point.

The number of instances in each variable are counted as well.

A picture containing text

Description automatically generated

Then if there are any duplicates present, which there are not, they are dropped. The data is also checked for any null values to ensure there aren’t any values that could affect the accuracy of the model.

Text

Description automatically generated

The value of ‘blueWins’ was double checked to ensure there are an even number of instances where the blue team won and game and lost a game. Since this will be the target variable for the model this helps the accuracy of the model.

Graphical user interface, text, application

Description automatically generated

Constructing and tuning the SVM Model

Once the dataset has been properly sanitised and pre-processed the model can be constructed. The SVM model was chosen due to its ability to create a classification model between two groups of data, creating a hyperplane between these two groups to be able to predict data. In this case the first group is “blueWins” which specifies whether the game was won or lost. The rest of the data specific to how well the team is performing in the first 10 minutes of the game is used to create a hyperplane between the models target data, “blueWins” and the rest of the dataset.

Text

Description automatically generated

First two variables called “X” and “y” should be specified, with X being the value of “blueWins”, and y being the rest of the dataset. X represents the target variable required to build a SVM classifier, and the y represents all other columns of the dataset.

Text

Description automatically generated

The dataset is then split up into three parts, with the training set holding 70% of the dataset, with the remaining 30% of the dataset being split between the validation and test set.

Next the SVM classifier is created using the “fit” function, with the training set passed through as the parameter for the model.

After this the model has been created, the “predict” function can be used to measure and print out the accuracy of the model we have created.

Text

Description automatically generated

The accuracy of the first SVM model created was 73%. This model should be tuned appropriately to see if the accuracy of the model could potentially be greater, as the current model has not been tuned at all.

For tuning the model, a GridSearchCV function was used to see if a more accurate model could be found, while also finding appropriate parameters and hyperparameters which can be used to create more accurate models using the validation and testing datasets which have not be classified yet.

This function takes in parameters and used to brute force through several SVM classifications to find the most accurate. However, this process is very time and performance sensitive, with this process taking as much as an hour to complete using the provided parameters.

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

The accuracy produced from the tuned model was actually smaller than the first model created, however this process provided the correct hyperparameters which can be used in a model which uses the validation and testing dataset which was split up before any SVM model was constructed.

Testing Results

One a tuned model has been created, the hyperparameters taken from the results of the GridSearchCV function can be used alongside the validation dataset which can be properly classified.

Text

Description automatically generated

This model produces an accuracy of 75%, the most accurate model produced so far.

Finally, the classification of the testing dataset is outputted, showing an accuracy of 73% which is slightly lower than the classification of the validation data set.

Graphical user interface, text, application

Description automatically generated

Discussion

Before any analysis or manipulation of data can even begin to start it must be properly pre-processed. Pre-Processing is responsible for keeping a dataset precise and valid. (Indeed, 2022)

In the case of the chosen dataset, very minimal processing was needed to get the most out of the dataset. Some columns were removed to keep the volume of data lower when constructing the model. Data was collected for the enemy team during a game of league of legends however the data that was chosen as the target variable had nothing to do with whether the red team won or not and was the only necessary pro-processing required to carry out the SVM classification of the dataset. Overall 75% is a successful accuracy for the model, and proves that the dataset has been successfully classified through the SVM algorithm, and would not have been achieved without proper pre-processing and tuning of the dataset.

Appendices

References

Indeed., 2022. What is data pre-processing? (with importance and examples) Available at : <https://ca.indeed.com/career-advice/career-development/data-preprocessing#:~:text=Importance%20of%20data%20preprocessing&text=It%20improves%20accuracy%20and%20reliability,It%20makes%20data%20consistent>.

McGregor, M., 2020. SVM Machine learning Tutorial – What is the support vector machine algorithm, Explained with code examples. Available at: <https://www.freecodecamp.org/news/svm-machine-learning-tutorial-what-is-the-support-vector-machine-algorithm-explained-with-code-examples/>