Applied Science and Technology Research Journal Volume 2 Number 2, November 2023
e- ISSN : 2963-6698
https://journal.upy.ac.id/index.php/ASTRO/index

Multi-Layer Perceptron Model for Dota 2 Game Results from UCI Using MLP Classifier

Galih Yudhistira*[1] , Pika Aliya Widiastuti[2] , Rahyuni[3] , Tri Hastono[4], Eko Harry Pratisto[5]
[1234] Informatics, Universitas PGRI Yogyakarta, Indonesia
[5]School of Management and Marketing, Curtin University, Australia

E-mail: **[1] galihyudhistira125@gmail.com** , [2]pikaaliya12345@gmail.com
[3]rahyunitaekwondo@gmail.com , [4]trihastono@upy.ac.id, [5]eko.harry@gmail.com

Abstract

Dota 2 is a genre game Moba in the PC (Personal Computer) system battle arena game online (online) with multiplayer (bringing together 2 players in 1 machine). Game Dota 2 consists of 2 opposing teams To get the victory, every team has 5 players who can choose hero 1 from 121 different heroes. Study This discusses the use of the Multi-Layer Perceptron (MLP) model to predict the results Dota 2 game. The author uses the UCI dataset containing historical data of Dota 2 matches, processed and trained with the MLP model using MLPClassifier from the scikit learn Python library. The data preprocessing process includes normalization features and handling of missing data. Training involves hyperparameter selection and validation cross To prevent overfitting. Although the MLP model is successful in predicting results with accuracy high, the author takes notes room For improvement, like additional features or the use of more models complex. In research, This obtained results with Accuracy Train results: 68.06%, Accuracy Test: 58.00%, Accuracy Precision: 58.53%, Accuracy Recall: 73.50%, Accuracy f1: 63.39%.

Keywords : *MLPClassifier, Multi-Layer Perceptron, Dota2*

1. INTRODUCTION

DOTA 2 is A game *multiplayer online battle arena (multiplayer game).* Multiplayer games ie games that can be played by two or more players at the same time and played online. The servers available in DOTA 2 have multiple servers based on location from each DOTA 2 player including Asia, Australia, India, Dubai, Russia, South Africa, Europe East, Europe West, US West, Chile, South America, US East, and Peru. DOTA 2 is played in two Teams containing five players each. Each team's headquarters center (Ancient) that became a decider wins Because the first team to destroy *Ancient* is the winner. In multiplayer games, players Can interact with all players, from the stronghold team or stronghold enemy so that there is compactness between players For winning matches [8].

In the gaming industry, Dota 2 has been one game competitive and very popular. Quality games from Dota 2 players are already reflected in the results match which are already achieved. In an effort to understand the influencing factors results Dota 2 game, already Lot of research has already been carried out in the field. The Dota 2 dataset from the UCI Machine Learning Repository has provided information related to Dota 2 matches and includes attributes of players as well as characteristics relevant to games. This dataset gives a chance for the researcher To dig outlook important factors that can influence results game and identify significant patterns.

In research For analyzing and predicting results, The DOTA 2 game uses the Multi-Layer Perceptron (MLP) Model. MLP is one of the types of network model nerve imitation already proven effective in various task modeling and classification. Perceptron is a technique network of existing nerves Lots are used in the classification and regression process. That matters because MLP has a level of accuracy and more Good in data mining Via studying this, we hope we can contribute To understanding more about the factors that play a role in the results Dota 2 game. Research results this is possible to use as a base For developing system recommendations or more game strategy intelligence in the Dota 2 community [3].

2. RESEARCH METHODS

2.1. Research Steps

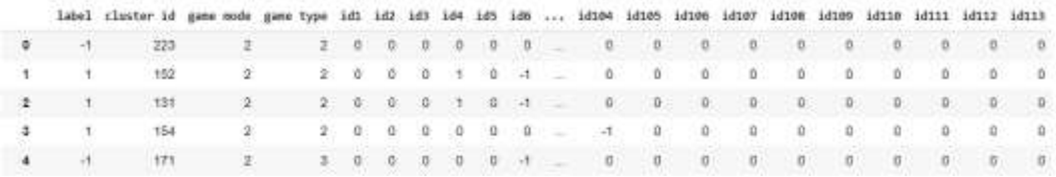
Steps possible analysis used in study This between other :

- Data Preparation
  Collect Dota 2 data from the UCI Machine Learning Repository. Import the required libraries such as pandas, numpy, and sklearn. Load data into something appropriate data structure with data frames using pandas.
- Data Exploration and Processing
  Do analysis data exploration to understand Available attributes and labels. Identification of attributes numerical and categorical and carrying out data processing such as coding category become numeric, normalization of attributes numeric, and processing missing values.
- Data Sharing
  Separate the data into training data and test data.
- Model Training
  Import MLPClassifier from sklearn. Define -model parameters such as amount layers, number of units per layer, and function Activation. Train the model using training data with the call fit method in MLPClassifier.
- Model Evaluation
  Use existing models trained To do predictions on test data. Count metric relevant evaluation like accuracy, precision, recall, and f1-score. Analysis results evaluation To evaluate model performance.
- Model Tuning
  Try variations of model parameters like amount layers, number of units per layer, or function Activation. Use technique validation cross (cross-validation) to get estimation more performance reliable.
- Prediction
  Once the model is considered adequate, use that model To make predictions on new data.

2.2. Method of collecting data

Data used in the study This is the data taken from the UCI Machine Learning Repository Dataset. the data can be taken from https://archive.ics.uci.edu/ml/datasets/Dota2+Games+Results

2.3. Research data



	label	cluster id	game mode	game type	id1	id2	id3	id4	id5	id6	...	id104	id105	id106	id107	id108	id109	id110	id111	id112	id113
0	-1	223	2	2	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	1	152	2	2	0	0	0	1	0	-1	...	0	0	0	0	0	0	0	0	0	0
2	1	131	2	2	0	0	0	1	0	-1	...	0	0	0	0	0	0	0	0	0	0
3	1	154	2	2	0	0	0	0	0	0	...	-1	0	0	0	0	0	0	0	0	0
4	-1	171	2	3	0	0	0	0	0	-1	...	0	0	0	0	0	0	0	0	0	0

Source data image: https://archive.ics.uci.edu/ml/datasets/Dota2+Games+Results

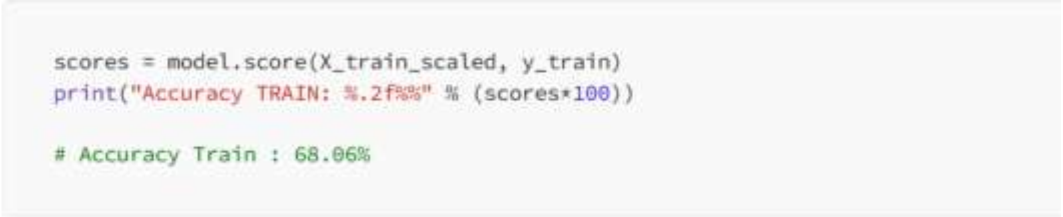
In this dataset containing the winning team match (1 or -1) as label, cluster id (location or server), game mode (all pick, single draft, captains mode, random draft, ability draft, least played, all random, limited heroes, captains draft, and all random deathmatch), type game (ranked all pick, captains mode, random draft), hero as column id1 id113 (1 indicates the selected hero For team, -1 indicates the hero chosen by the enemy team).

On scaling feature. scalar is a function used To mark something feature in data collection, done For the entire training (x) and test (x) sets. How to normalize feature or input variables viz use min-max scaling like Can seen under This :

$$x_{scaled} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

3. RESULTS AND DISCUSSION

The dataset used in this research comes from the UCI Machine Learning Repository and includes various features related to Dota2 game results, such as hero selection, game time, and player performance statistics. This research aims to use the MLPClassifier from the sci-kit-learn library to train a Multi-Layer Perceptron (MLP) model.

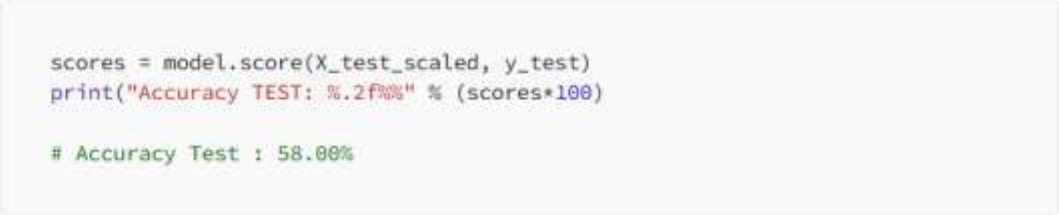


```
scores = model.score(X_train_scaled, y_train)
print("Accuracy TRAIN: %.2f%%" % (scores*100))

# Accuracy Train : 68.06%
```

Figure 1. Train Data Accuracy

'scores = model. score (x_train_scaled, y_train)' is used For counting model accuracy on training data '(x_train_scaled and y_train)'. The function ' score( )' is something available methods _ in working model object count accuracy. that method compares predictions produced by the model with the actual value in the training data. Next, ' print("Accuracy TRAIN: %.2f%%" % (scores*100))' works For print mark accuracy on training data to in form percentage. Use '%2f%%', to combine mark accuracy with the string "Accuracy TRAIN: " and convert the mark the become form percentage. The final result will print mark accuracy on training data in percentage format, on the data results in "TRAIN Accuracy: 68.06%".

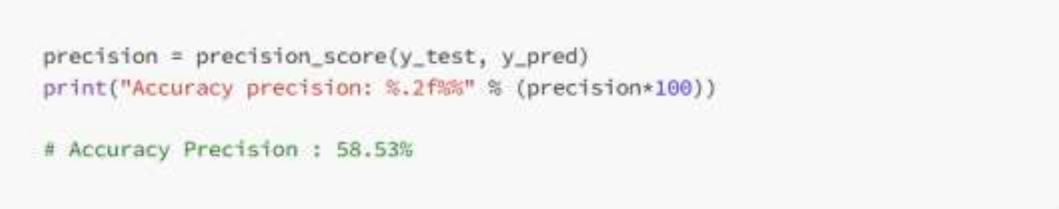


```
scores = model.score(X_test_scaled, y_test)
print("Accuracy TEST: %.2f%%" % (scores*100)

# Accuracy Test : 58.00%
```

Figure 2. Test Data Accuracy

'scores = model. score (x_test_scaled, y_test)' is used For count model accuracy on test data '(x_test_scaled and y_test)'. Function score( ) is something available methods _ in working model object count accuracy. that method compares predictions produced by the model with the actual value in the test data. Next, ' print("Accuracy TRAIN: %.2f%%" % (scores*100))' works For print mark accuracy on training data to in form percentage. Using '%2f%%', combine mark accuracy with the string "Accuracy TEST: " then convert mark the to become form percentage. The final result will print mark accuracy on test data in percentage format, on the data results in "Accuracy TEST: 58.00%"

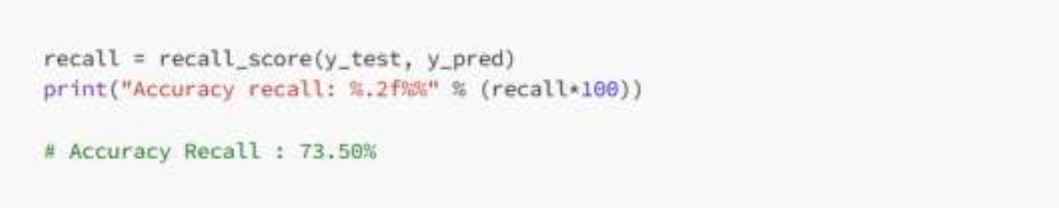


```
precision = precision_score(y_test, y_pred)
print("Accuracy precision: %.2f%%" % (precision*100))

# Accuracy Precision : 58.53%
```

Figure 3. Precision Data Accuracy

'precision = precision_ score (y_test, y_pred)' is used For count precision value or level accuracy predictions positive by the model. The function ' precision_ score ( ) ' is one existing method _ in the module sklearn. metrics For calculating precision. Precision works measure the extent of the predictions positives produced by the model are correct or appropriate. Next, ' print("Accuracy precision: %.2f%%" % (precision*100))' is used For print precision value to in form percentage. With using '%.2f%%', we combine precision value with the string "Accuracy precision: " and convert the mark to the form percentage using two number decimals. The final printed result is precision value in percentage format and obtained result"Accuracy precision: 58.53%". The precision value or level accuracy predictions are positive is 58.53%. This matter means from all predictions positives are created by the model, about 58.53% of them Correct or appropriate.

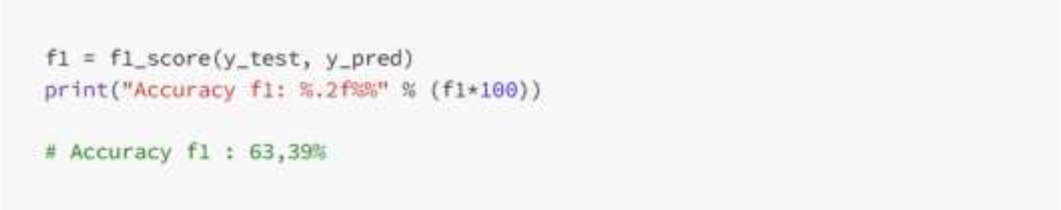


```
recall = recall_score(y_test, y_pred)
print("Accuracy recall: %.2f%%" % (recall*100))

# Accuracy Recall : 73.50%
```

Figure 4. Data Recall Accuracy

' recall = recall_score (y_test, y_pred)' is used To count recall value or level the model's ability to detect in a way true positive instances in the test data. The function ' recall_score ( )' is available methods _ in the module sklearn. metrics For count recalls. Recall works measure to what extent the model is capable of finding or detecting positive instances with Correct. Next, ' print("Accuracy recall: %.2f%%" % (recall*100))' works For print inner recall value form percentage. Using '%.2f%%', to combine recall value with the string "Accuracy recall: " then convert mark the become form percentage use two number decimal. The final result will print the recall value in percentage format, and the obtained result "Accuracy recall: 73.50%". The model's ability to detect positive instances with Correct namely 73.50%. That matter means about 73.50% of all positive instances in the test data are successfully detected by the model.



```
f1 = f1_score(y_test, y_pred)
print("Accuracy f1: %.2f%%" % (f1*100))

# Accuracy f1 : 63,39%
```

Figure 5. F1 Data Accuracy

'f1_score' is used For count intermediate F1 score y_test and y_pred . The F1 score is the size used _ To combine the precision and recall of a classification model. After the F1 score is calculated, the result is displayed using the 'print' function With using '%.2f%%', we arrange the appearance number decimal two places at the back coma as well as add sign percent behind the printed output namely "F1 Score: 63.39%".

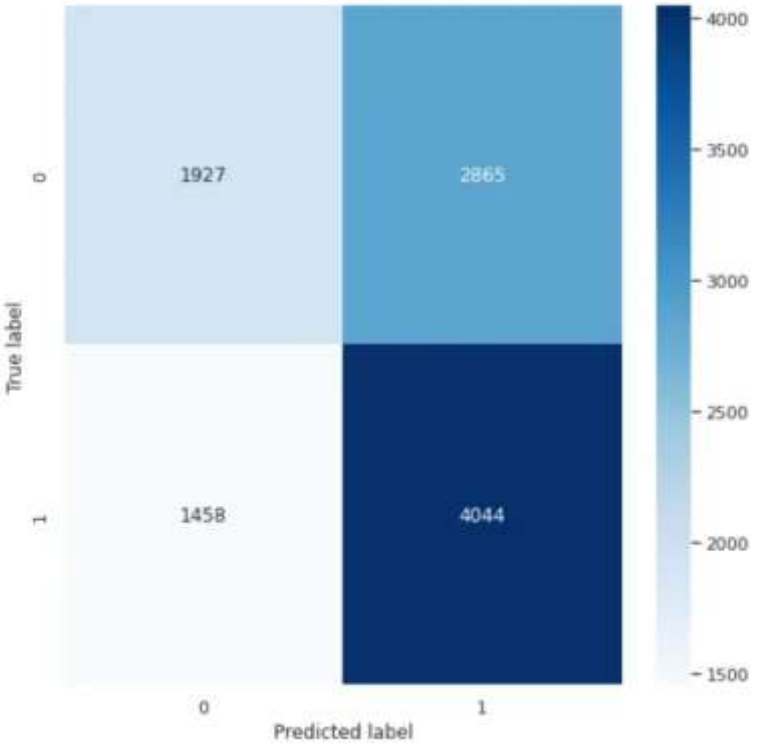


Figure 6. Value table

On the chart used predict marks from the label from each label.

CONCLUSION

Use MLPClassifier To predict results proven Dota 2 game effective For do classification results game Dota 2. By using a dataset from UCI, we can train a model to recognize patterns and produce more predictions _ accurate related results match with Accuracy Train results: 68.06%, Accuracy Test: 58.00%, Accuracy Precision: 58.53%, Accuracy Recall: 73.50%, Accuracy f1: 63.39%.

REFERENCES

[1] Irfan, Muhammad, et al. “Sistem Klasifikasi Kendaraan Berbasis Pengolahan Citra Digital Dengan Metode Multilayer Perceptron.” *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, vol. 7, no. 2, 2017, p. 139, https://doi.org/10.22146/ijeis.18260.

[2] Liyantoko, Apri Nur, et al. “Klasifikasi Sel Darah Putih Dan Sel Limfoblas Menggunakan Metode Multilayer Perceptron Backpropagation.” *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, vol. 9, no. 2, 2019, p. 173, https://doi.org/10.22146/ijeis.49943.

[3] Metode, Menggunakan, and M. L. P. Multi. *Klasifikasi Pada Dataset UCI Nursery Dengan*. 1980.

[4] Mlp, Perceptron, et al. *PENGGUNAAN SERTIFIKAT VAKSIN MENGGUNAKAN METODE MULTI-LAYER FAST TEXT Skripsi Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar*. 2023.

[5] Purwaningsih, Nunik. “Penerapan Multilayer Perceptron Untuk Klasifikasi Jenis Kulit Sapi Tersamak.” *Jurnal TEKNOIF*, vol. 4, no. 1, 2016, pp. 1–7, https://ejournal.itp.ac.id/index.php/tinformatika/issue/view/64.

[6] Putra, I. Gede Rusdy Mahayana, et al. “Identifikasi Citra Ukiran Ornamen Tradisional Bali Dengan Metode Multilayer Perceptron.” *SINTECH (Science and Information Technology) Journal*, vol. 4, no. 1, 2021, pp. 29–39, https://doi.org/10.31598/sintechjournal.v4i1.552.

[7] Ritonga, Alven Safik, and Suryo Atmojo. “Pengembangan Model Jaringan Syaraf Tiruan Untuk Memprediksi Jumlah Mahasiswa Baru Di PTS Surabaya (Studi Kasus Universitas Wijaya Putra).” *Jurnal Ilmiah Teknologi Informasi Asia*, vol. 12, no. 1, 2018, p. 15, https://doi.org/10.32815/jitika.v12i1.213.

[8] Sulistyani, Arum Rizqi. *DOTA 2 Yang Berlangsung Via*. pp. 1–13.

[9] Yanto, Musli, et al. “Jaringan Syaraf Tiruan Perceptron Untuk Penentuan Pola Sistem Irigasi Lahan Pertanian Di Kabupaten Pesisir Selatan Sumatra Barat.” *Sebatik*, vol. 22, no. 2, 2018, pp. 111–15, https://doi.org/10.46984/sebatik.v22i2.317.