UNIWERSYTET IM. ADAMA MICKIEWICZA W POZNANIU
WYDZIAŁ NAUK SPOŁECZNYCH
KOGNITYWISTYKA

Gracjan Popiółkowski

Raport skuteczności przewidywań różnych modeli uczenia maszynowego

Effectiveness Report of Predictions From Various Machine Learning Models

Poznań 2023

Spis treści

Wprowadzenie

W tej pracy zaprezentuje wyniki uczenia sztucznej sieci neuronowej z różnymi parametrami, którą wykorzystam do prognozowania wyniku meczu na podstawie danych z pierwszych 15 minut popularnej gry "League of Legends". Pierwszy kwadrans rozgrywki stanowi istotny moment meczu, dlatego stawiam sobie pytania: Jak dużą precyzją możemy przewidzieć rezultat meczu, bazując na kluczowych statystykach obu drużyn? Czy pierwsze 15 minut są decydujące?

ROZDZIAŁ 1

Prezentacja danych

1.1. Zbiór danych

Korzystam z zestawu danych "League of Legends Diamond Games (First 15 Minutes)", udostępnionego przez użytkownika Bena Fattoriego na platformie Kaggle. Dostęp do tego zbioru został umożliwiony przez twórców gry - Riot Games. Dane zostały zebrane z prawie 50000 meczów graczy rangi diamentowej (jednej z najwyższych) na serwerze "NA1" i zawierają kluczowe statystyki dotyczące przebiegu rozgrywki z pierwszych 15 minut gry.

1.2. Opis danych

Spośród 19 kolumn danych, przed dalszą analizą usunięto 4 z nich:

- `id` (numer danych) - mogłoby to zaburzyć pracę algorytmów ML.
- `matchId` (indywidualny identyfikator meczu) - nie ma wpływu na rozgrywkę.
- `blueDragonKills` (ile smoków zabiła drużyna niebieska)
- `redDragonKills` (ile smoków zabiła drużyna czerwona) - wartość zmiennej w obu drużynach w każdym meczu wynosi 0, więc zmienne nie mają znaczenia.

Dalsza analiza obejmuje 15 zmiennych, które są opisane poniżej:

- `blue_win` - zmienna kategoryzująca, przyjmująca wartości 0 i 1. 1 w przypadku, kiedy drużyna niebieska wygrała i 0, kiedy wygrała drużyna czerwona.

Pozostałe kolumny to zmienne numeryczne:

- `blueGold` - suma monet zdobytych przez drużynę niebieską, składającą się z 5 graczy.
- `redGold` - to samo odnośnie drużyny czerwonej.
- `blueMinionsKilled` - liczba potworów zabitych przez drużynę niebieską.
- `redMinionsKilled` - to samo odnośnie drużyny czerwonej.
- `blueJungleMinionsKilled` - liczba potworów zabitych przez *leśnika* drużyny niebieskiej.
- `redJungleMinionsKilled` - to samo odnośnie drużyny czerwonej.
- `blueAvgLevel` - średni poziom doświadczenia bohaterów graczy z drużyny niebieskiej w 15 minucie gry.
- `redAvgLevel` - to samo odnośnie drużyny czerwonej.
- `blueHeraldKills` - liczba potworów o nazwie "Herald" zabitych przez drużynę niebieską.

- `redHeraldKills` - to samo odnośnie drużyny czerwonej.
- `blueTowersDestroyed` - liczba zniszczonych wież drużyny niebieskiej.
- `redTowersDestroyed` - to samo odnośnie drużyny czerwonej.
- `blueChampKills` - ile razy łącznie drużynie niebieskiej udało się zabić bohatera z drużyny przeciwnej.
- `redChampKills` - to samo odnośnie drużyny czerwonej.

1.3. Przygotowanie danych

Tak jak wyżej pisałem 4 kolumny danych zostały usunięte. Zmienną kategoryzującą jest kolumna `blue_win`. Dodatkowo wykonałem zabieg standaryzacji danych na wszystkich 14 numerycznych cechach, czyli przekształciłem je w taki sposób, aby miały średnią wartość równą 0 i odchylenie standardowe równe 1. Może to poprawić efektywność użytych przeze mnie algorytmów uczenia maszynowego.

ROZDZIAŁ 2

Trzy algorytmy uczenia maszynowego

Problemem, którym się zajmujemy, polega na określeniu, która drużyna wygra. Z tego powodu będziemy pracować na zmiennej kategoryzacji `blue_win`. Wybrałem trzy algorytmy sztucznej inteligencji, które radzą sobie z kategoryzacją binarną:

1. `SVM` (Support Vector Machines - Maszyna wektorów nośnych) - to algorytm uczenia maszynowego używany zarówno do klasyfikacji, jak i regresji. Jego celem jest znalezienie hiperpłaszczyzny separującej dane w przestrzeni cech. Działa na zasadzie maksymalizacji odległości między najbliższymi punktami różnych klas (wektorami nośnymi).

2. `Sztuczna sieć neuronowa` - to model inspirowany strukturą mózgu, zbudowany z jednostek nazywanych neuronami, połączonych ze sobą. Neurony są zorganizowane w warstwy, a każde połączenie między nimi ma wagę. Proces uczenia polega na dostosowywaniu wag, aby model lepiej przewidywał wyniki.

3. `Random Forest` - algorytm oparty na idei tworzenia wielu drzew decyzyjnych. Każde drzewo jest trenowane na innym podzbiorze danych, a następnie głosowanie lub uśrednianie wyników drzew pozwala uzyskać stabilniejszą prognozę. Działa dobrze w przypadku dużej ilości cech. Algorytm ten jest uważany za bardzo skuteczny w przypadku klasyfikacji binarnej, co sprawia, że mam wobec niego duże oczekiwania.

W dalszej części tekstu przedstawię porównanie wyników z zastosowania tych 3 algorytmów.

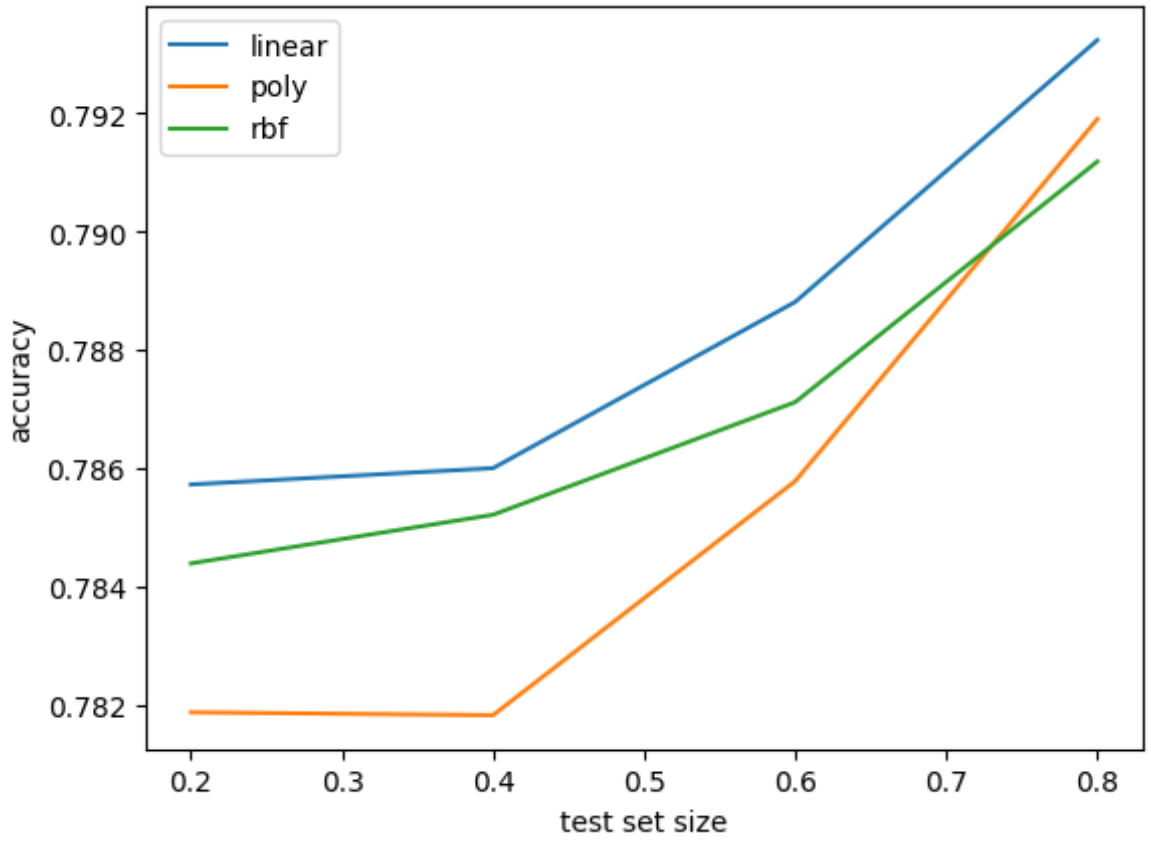
ROZDZIAŁ 3

Analiza wyników

3.1. SVM

3.1.1. Rozmiar zbioru treningowego i funkcja jądra

W algorytmie SVM sprawdziłem skuteczność klasyfikacji dla najróżniejszych wartości współczynników: Po pierwsze zrobiłem test skuteczności w przewidywaniu wyniku meczu dla kombinacji czterech wielkości proporcji zbioru treningowego do testowego (20% do 80%, 40% do 60%, 60% do 40% i 80% do 20%) z różnymi rodzajami funkcji jądra: liniową (linear), wielomianową (poly) i radialną (rdf).



Rysunek 3.1: Wykres przedstawiający skuteczność kategoryzacji dla kombinacji proporcji zbioru treningowego do testowego z rodzajami funkcji jądra.

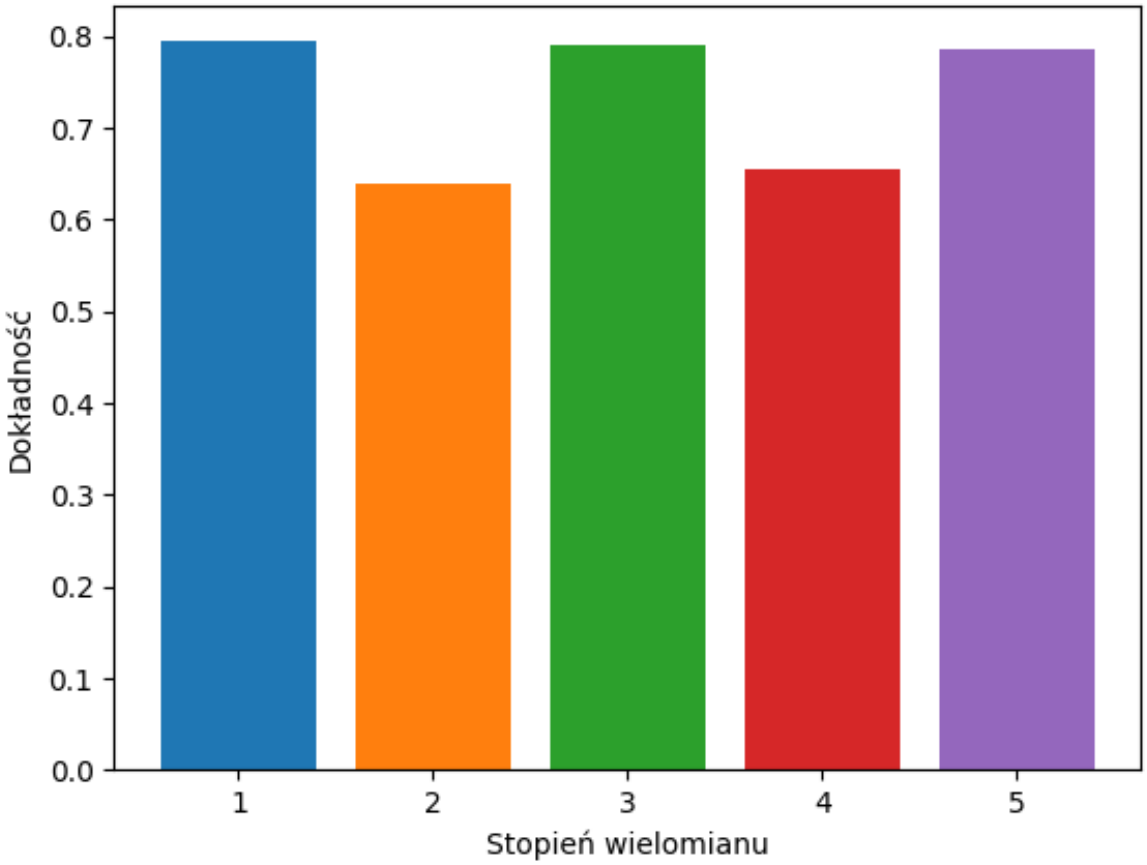
Na powyższym rysunku (3.1) widzimy, że największą skuteczność ma kombinacja o wielkości zbioru treningowego 80% i funkcji jądra liniowej. Skuteczność przewidzenia wyniku meczu dla tej kombinacji plasuje się na poziomie 79,32

3.1.2. Wartość parametru `c_value`

Następne co sprawdziłem to skuteczność z różnym parametrem `c_value`, jest to parametr, który kontroluje kompromis między osiągnięciem niskiego błędu treningowego a minimalizacją norm wag. Przy niskiej wartości `c_value` model jest bardziej tolerancyjny wobec błędów treningowych, a przy wysokiej wartości model silnie kara za błędy treningowe, co skuje dokładniejszym dopasowaniem punktów treningowych, ale może prowadzić to do przeuczenia sieci, sytuacji, w której sieć tak dokładnie przeanalizowała zbiór treningowy, że nie będzie sobie radziła z nowymi danymi. Wartości `c_value` jakie sprawdziłem to 0.001, 0.1 i 1. Najlepszy wynik uzyskałem dla wartości środkowej 0.1, dlatego nie sprawdzałem już więcej możliwości. Przy takim `c_value` = 0.1, dokładność przewidywania wynosi 79.34%.

3.1.3. Stopień wielomianu funkcji jądra

Ostatnie co sprawdziłem to stopień wielomianu używanego w funkcji jądra. Dokładność dla stopni od 1 do 5 i najwyższa skuteczność wyszła przy stopniu pierwszym równa 79,4%.



Rysunek 3.2: Wykres przedstawiający dokładność przewidzenia wyniku meczu dla różnych stopni wielomianu funkcji jądra.

3.1.4. Podsumowanie

Podsumowując, najlepsza kombinacja parametrów algorytmu SVM to:

- `Funkcja jądra`: liniowa,
- `Wielkość zbioru treningowego`: 80% danych,
- `c_value` = 0.1,
- `Stopień wielomianu funkcji jądra` = 1.

i taka kombinacja daje dokładność klasyfikacji na poziomie 79,4%.

3.2. Sztuczna sieć neuronowa

W moim algorytmie Sztucznych Sieci Neuronowych wielkość zbioru treningowego również ustawiłem na 80%, a testowego na 20%. Pierwszą warstwę gęstą skonfigurowałem tak, aby miała 16 neuronów i funkcję aktywacyjną `ReLU`. Drugą warstwę gęstą wyposażyłem w 8 neuronów, również z funkcją aktywacyjną `ReLU`, natomiast warstwę wyjściową jednym neuronem i funkcją aktywacji `sigmoid`. Przy 15 epokach treningowych model osiągnął skuteczność na poziomie 78,9%.

3.3. Random Forest

Dla modelu Random Forest również skorzystałem z wielkości zbioru treningowego równą 80%, a testowego - 20%. Po przetestowaniu różnych kombinacji parametrów, takich jak `n` (liczba drzew decyzyjnych) i `random_stat` (wartość ziarna losowości), najlepszy wynik, równy 78,58% skuteczności, został osiągnięty przy `n` = 100 oraz `random_stat` = 42. Ostatecznie wyniki modelu Random Forest nie okazały się lepsze od dobrze skonfigurowanych SVM i Sztucznej Sieci Neuronowej.

Podsumowanie

Reasumując, w mojej pracy użyłem trzech algorytmów uczenia maszynowego do przewidywania wyników meczów w grze "League of Legends", wykorzystując dane z pierwszych 15 minut gry. Otrzymane wyniki przewidywania dla poszczególnych algorytmów przy odpowiedniej konfiguracji to:

- `SVM` = 79,4%,
- `Sztuczna Sieć Neuronowa` = 78,9%,
- `Random Forest` = 78,58%.

Wnioskiem z analizy jest, że najlepiej poradził sobie odpowiednio skonfigurowany algorytm SVM. Jednak żaden z algorytmów nie osiągnął skuteczności powyżej 80%, co sugeruje, że na podstawie pierwszych 15 minut meczu trudno jest przewidzieć wynik z dokładnością większą niż 80%. Być może przekroczenie tej granicy wymagałoby dodania kolejnych cech do zbioru danych. Jednakże 14 cech, które zostały uwzględnione, to najważniejsze parametry rozgrywki, na które gracz może wpłynąć w pierwszych 15 minutach gry. Warto zauważyć, że cały mecz w grze "League of Legends" trwa średnio od 20 do 50 minut, więc osiągnięta skuteczność na poziomie 80% jest już wysoka.