Projekt: Eksploracja Danych

Etap Drugi: Przygotowanie Danych + Modelowanie

Statystyki Wczesnej Fazy Gry w League of Legends

AUTOR: MICHAŁ TARNOWSKI
IDENTYFIKATOR STUDENTA: 193324
DATA: 10.06.2025

Opis wejściowego zbioru danych

Zbiór danych do tej analizy składa się z **9879** rekordów z meczów *League of Legends* na wysokim poziomie rywalizacji.

- **Pochodzenie:**
  Dane zostały pozyskane z oficjalnego API firmy Riot Games.

- **Format:**
  CSV

- **Liczba rekordów:**
  9879

- **Liczba cech (kolumn):**
  40

Dane przedstawiają kompleksowy obraz stanu gry po pierwszych 10 minutach – kluczowego okresu znanego jako „wczesna faza gry". Zbiór zawiera cechy numeryczne, szczegółowo opisujące różne metryki wydajności dla obu drużyn: „niebieskiej" i „czerwonej". Metryki te obejmują wskaźniki ekonomiczne (złoto, doświadczenie), statystyki bojowe (zabójstwa, zgony, asysty) oraz kontrolę celów (smoki, Heroldy, wieże).

- **Jakość i charakterystyka danych:**
  Kluczową cechą tego zbioru danych, potwierdzoną podczas wstępnej eksploracji, jest jego kompletność; **nie ma brakujących wartości**, co eliminuje potrzebę imputacji. Ponadto, zmienna docelowa, **`blueWins`**, jest wyjątkowo dobrze zbalansowana, z drużyną niebieską wygrywającą w 4930 grach (49,9%) i przegrywającą w 4949 grach (50,1%).

  Ta równowaga jest bardzo korzystna dla uczenia maszynowego, ponieważ zapobiega tworzeniu przez model tendencyjności w kierunku klasy większościowej i pozwala na bardziej sprawiedliwą i dokładną ocenę jego mocy predykcyjnej.

Cel(e) eksploracji danych i kryteria sukcesu

1. **Cel główny (predykcyjny):**
   Skonstruowanie i ocena modelu uczenia maszynowego zdolnego do przewidywania zwycięzcy meczu (`blueWins`) przy użyciu tylko danych dostępnych w 10 minucie gry.

2. **Cel drugorzędny (interpretacyjny):**
   Stworzenie modelu, który będzie wysoce interpretowalny. Celem jest nie tylko przewidywanie, czy drużyna wygra, ale zrozumienie, *dlaczego*. Obejmuje to identyfikację konkretnych statystyk w grze, które są najsilniejszymi predyktorami zwycięstwa, co pozwala na ilościowe zweryfikowanie i rozszerzenie wniosków uzyskanych podczas wstępnej eksploracji danych.

3. **Kryteria sukcesu:**

   1. Osiągnięcie dokładności (**`accuracy`**) modelu predykcyjnego, która znacząco przewyższa próg losowy (**50%**). Za sukces uznaje się wynik przekraczający **70%**.
   2. Zidentyfikowanie i ilościowe określenie istotności cech, które w największym stopniu wpływają na wynik meczu.

Omówienie dalszych kroków do podjęcia

1. **Wybór zadania eksploracji danych:**
   Problem stanowi klasyczne zadanie klasyfikacji binarnej. Celem modelu jest przypisanie każdej grze (każdemu wierszowi) do jednej z dwóch dyskretnych klas: **1**, oznaczającej zwycięstwo drużyny niebieskiej, lub **0**, oznaczającej przegraną.

2. **Wybór algorytmu eksploracji danych:**
   `DecisionTreeClassifier` został wybrany jako główny algorytm do tego zadania. Model ten został wybrany ze względu na swoją naturę „*białej skrzynki*", co oznacza, że jego wewnętrzna logika jest przejrzysta i może być łatwo wizualizowana. Bezpośrednio wspiera to cel projektu, jakim jest interpretowalność, ponieważ pozwala nam śledzić dokładne ścieżki decyzyjne, których używa model, oraz hierarchicznie porządkować cechy według ich ważności.

3. **Wybór metody oceny wyników:**
   Pojedynczy podział na zbiór treningowy i testowy może dawać wyniki zależne od konkretnego losowego podziału danych. Aby uzyskać bardziej solidną i uogólnioną miarę prawdziwej wydajności modelu, zastosowano 10-krotną walidację krzyżową. Technika ta polega na trenowaniu i walidacji modelu 10 razy na różnych podzbiorach danych i uśrednianiu wyników. Proces ten zapewnia bardziej stabilne oszacowanie tego, jak model zachowałby się na nowych, niewidzianych danych. Ocena skupiała się na kluczowych metrykach klasyfikacji: dokładności (**`accuracy`**), precyzji (**`precision`**) i czułości (**`recall`**).

Przygotowanie danych

- **Brakujące dane i ujednolicenie:**
  Początkowe skanowanie danych potwierdziło wysoką jakość zbioru danych, ponieważ **nie było brakujących wartości**. Usprawniło to fazę przygotowania i wyeliminowało potrzebę stosowania technik imputacji danych.

- **Selekcja podzbioru cech:**
  Początkowy zbiór danych zawierał **40 cech**. Aby zbudować bardziej skoncentrowany i wydajny model, a także uniknąć problemu współliniowości, zestaw ten został zredukowany do **21 cech** poprzez następujące kroki:

  1. **Usunięcie identyfikatora:**
     Kolumna **`gameId`** została usunięta, ponieważ jest unikalnym identyfikatorem dla każdego meczu i nie posiada żadnej wartości predykcyjnej.

```
29  # Drop the gameId as it is just an identifier
30  if 'gameId' in df.columns:
31      df = df.drop('gameId', axis=1)
32      print("Dropped 'gameId' column.")
```

  2. **Inżynieria cech i redukcja redundancji:**
     Zamiast analizować statystyki obu drużyn oddzielnie, stworzono cechy "różnicowe" (np. **`blueGoldDiff`**, **`blueExperienceDiff`**). Te cechy bezpośrednio mierzą względną przewagę jednej drużyny nad drugą. W konsekwencji, większość indywidualnych statystyk dla drużyny czerwonej (np. **`redTotalGold`**, **`redKills`**) została usunięta, ponieważ informacja w nich zawarta jest już efektywnie reprezentowana przez cechy różnicowe. Ten krok znacząco redukuje współliniowość i upraszcza model.

```python
# We will keep the blue team's primary stats and the diff stats.
features_to_drop = [
    'redWardsPlaced', 'redWardsDestroyed', 'redFirstBlood', 'redKills', 'redDeaths',
    'redAssists', 'redEliteMonsters', 'redDragons', 'redHeralds', 'redTowersDestroyed',
    'redTotalGold', 'redAvgLevel', 'redTotalExperience', 'redTotalMinionsKilled',
    'redTotalJungleMinionsKilled', 'redGoldPerMin', 'redCSPerMin'
]

# Create the feature set X and target y
X = df.drop([TARGET] + features_to_drop, axis=1)
y = df[TARGET]
```

3. **Finalny zbiór cech:**
   Ostateczny model został wytrenowany na wyselekcjonowanym zestawie **21** cech, skupiając się głównie na statystykach drużyny niebieskiej oraz kluczowych, metrykach różnicowych, które zostały poddane agregacji.

```
Number of features: 21
Final features used for modeling:
['blueWardsPlaced', 'blueWardsDestroyed', 'blueFirstBlood', 'blueKills', 'blueDeaths', 'blueAssists', 'blueEliteMonsters',
 'blueDragons', 'blueHeralds', 'blueTowersDestroyed', 'blueTotalGold', 'blueAvgLevel', 'blueTotalExperience',
 'blueTotalMinionsKilled', 'blueTotalJungleMinionsKilled', 'blueGoldDiff', 'blueExperienceDiff', 'blueCSPerMin',
 'blueGoldPerMin', 'redGoldDiff', 'redExperienceDiff']
```

- **Transformacja i Standaryzacja Danych:**
  Zbiór danych zawiera wyłącznie cechy numeryczne. Wybrany algorytm Drzewa Decyzyjnego operuje na zasadzie podziałów opartych na regułach (np. "`gold > 5000`"), przez co **nie jest wrażliwy na skalę wartości wejściowych**. Dlatego techniki skalowania cech, takie jak standaryzacja czy normalizacja, nie były konieczne dla tego projektu i nie zostały zastosowane.

Tworzenie modelu

Wybór parametrów algorytmu

Klasyfikator **`DecisionTreeClassifier`** został skonfigurowany z parametrem **`max_depth=4`**. Ten hiperparametr działa jako mechanizm wstępnego przycinania, ograniczając wzrost drzewa do czterech poziomów.

Był to świadomy wybór, aby zapobiec nadmiernej złożoności modelu i „zapamiętywaniu" danych treningowych (zjawisko znane jako *overfitting*), co zagroziłoby jego zdolności do generalizacji na nowe gry. Parametr

`random_state` został ustawiony na **`42`**, aby zapewnić powtarzalność wyników, gdzie liczba ta jest też nieoficjalnym, przyjętym standardem.

```
7  dt_model = DecisionTreeClassifier(max_depth=4, random_state=42)
```

Analiza wynikowego modelu

- **Ważność cech:**
  Wytrenowany model zapewnił wyraźną i klarowną hierarchię ważności cech. Analiza wykazała, że przewidywania modelu są w przeważającej mierze zdominowane przez przewagę ekonomiczną.

  Cechy **`blueGoldDiff`** i **`redGoldDiff`** (które są swoimi idealnymi odwrotnościami) zostały zidentyfikowane jako najbardziej krytyczne, wspólnie odpowiadając za **ponad 94% wagi decyzyjnej modelu**.

```
--- Feature Importances ---
redGoldDiff                 0.835789
blueGoldDiff                0.106948
```

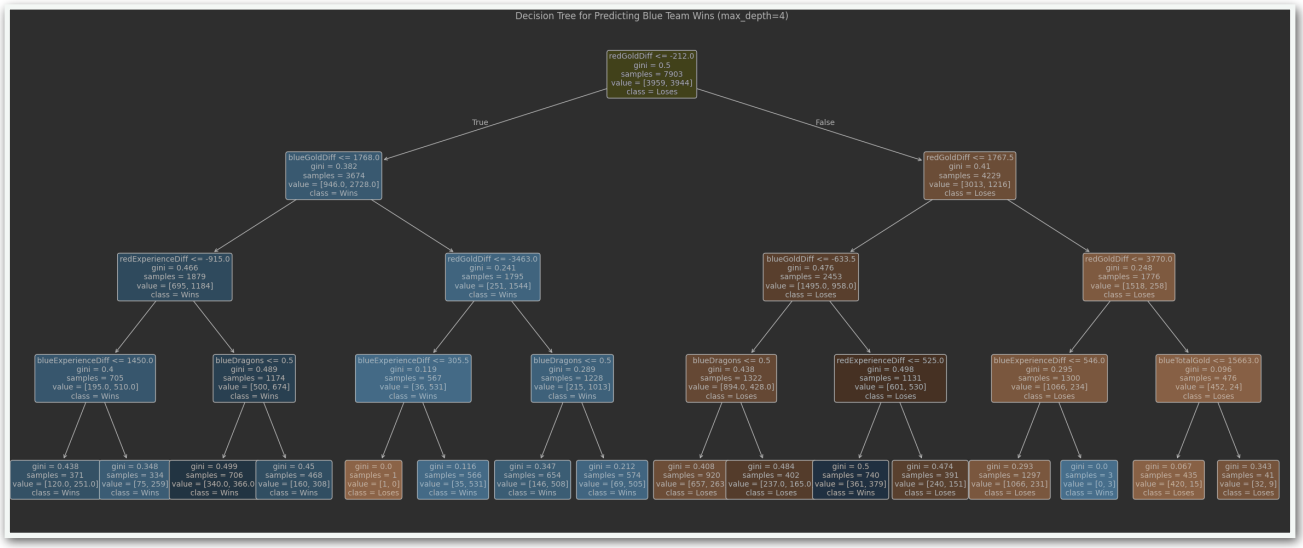
  Kolejne najważniejsze cechy, **`redExperienceDiff`** i **`blueDragons`**, miały stosunkowo minimalny wpływ. Odkrycie to silnie sugeruje, że na tym poziomie rozgrywki przewaga w złocie po 10 minutach jest najbardziej decydującym czynnikiem w przewidywaniu zwycięzcy.

```
redExperienceDiff           0.024184
blueDragons                 0.022793
```

- **Analiza ścieżki decyzyjnej:**
  Wizualna reprezentacja drzewa ilustruje tę logikę w działaniu. **`Root`** dzieli dane na podstawie różnicy w złocie. Na przykład, gry, w których drużyna niebieska ma znaczną przewagę w złocie, są natychmiast kierowane ścieżką z wysokim prawdopodobieństwem klasyfikacji „Wygrana”.

  Kolejne węzły dalej precyzują te przewidywania, wykorzystując różnicę w doświadczeniu i, w niektórych przypadkach, liczbę zdobytych smoków. Pokazuje to, że choć złoto jest najważniejsze, drugorzędne cele mogą nadal wpływać na wynik w bardziej wyrównanych grach.

Drzewo decyzyjne dla przewidywania wygranej drużyny



„Kolejne węzły dalej precyzują te przewidywania, wykorzystując różnicę w doświadczeniu i, w niektórych przypadkach, liczbę zdobytych smoków."

Ocena wyników za pomocą wybranej metody

10-krotna walidacja krzyżowa zapewniła rzetelną ocenę wydajności modelu na niewidzianych danych.

```
--- Performing 10-Fold Cross-Validation ---

Average Accuracy: 0.7222 (+- 0.0117)
Average Precision: 0.7315 (+- 0.0162)
Average Recall (Sensitivity): 0.7020 (+- 0.0385)
Average F1-Score: 0.7157 (+- 0.0175)
```

- **Średnia dokładność (Accuracy):**
  72,2% (±1,2%). Model poprawnie przewiduje zwycięzcę w około 72 na każde 100 gier.

- **Średnia precyzja (Precision):**
  73,2% (±1,6%). Kiedy model przewiduje, że drużyna niebieska wygra, jest to poprawne w 73,2% przypadków. Wskazuje to na dobry poziom

wiarygodności dla pozytywnych przewidywań.

- **Średnia czułość (Recall):**
  70,2% (±3,9%). Model z powodzeniem identyfikuje 70,2% wszystkich gier, które drużyna niebieska faktycznie wygrała. Pokazuje to, że jest zdolny do wychwycenia większości scenariuszy zwycięskich.

Ogólnie rzecz biorąc, te metryki dowodzą, że prosty, interpretowalny model może osiągnąć solidne wyniki predykcyjne, stanowiąc cenne narzędzie do zrozumienia dynamiki gry.

Podsumowanie i wnioski

Krótki przegląd procesu

Projekt ten z powodzeniem przeszedł od analizy eksploracyjnej do modelowania predykcyjnego. Zbudowano, wytrenowano i oceniono model Drzewa Decyzyjnego do przewidywania wyników meczów w League of Legends na podstawie danych z pierwszych 10 minut. Proces ten obejmował staranne przygotowanie danych, w tym strategiczny wybór cech, oraz rygorystyczną ocenę za pomocą 10-krotnej walidacji krzyżowej.

Stopień pokrycia celów

Cele projektu zostały w pełni osiągnięte:

1. **Cel predykcyjny:**
   Zbudowany model osiągnął średnią dokładność na poziomie **72,2%**, co wyraźnie przekracza zdefiniowane kryterium sukcesu (70%) i stanowi solidny wynik predykcyjny.

2. **Cel interpretacyjny:**
   Został w pełni zrealizowany. Dzięki zastosowaniu modelu "białej skrzynki" (Drzewo Decyzyjne) udało się jednoznacznie zidentyfikować kluczowe czynniki decydujące o zwycięstwie. Analiza ważności cech niepodważalnie wykazała, że **różnica w złocie (`blueGoldDiff`) jest dominującym predyktorem**, odpowiadając za ponad 94% wagi decyzyjnej modelu. Potwierdza to, że abstrakcyjne pojęcie „przewagi ekonomicznej” może być bezpośrednio przetłumaczone na potężną cechę predykcyjną.

3. **Wniosek końcowy:**
   Analiza potwierdziła z rygorem statystycznym, że pierwsze 10 minut meczu w League of Legends na wysokim poziomie rozgrywek to nie tylko faza wstępna, ale często okres, który w decydujący sposób kształtuje ostateczny wynik. Model posłużył jako efektywne narzędzie analityczne do weryfikacji tej hipotezy.