בס"ד

חיזוי תוצאות משחקי כדורגל בעזרת למידת מכונה

אילת ג'יבלי (208691675), עדי מלאכי (207740846)

הקדמה

הרעיון לפרויקט זה הגיע בעקבות אתר ווינר - אתר הימורי ספורט ישראלי באחריות הטוטו. במסגרת תוכניות ההימורים צריך לנחש תוצאות של משחקי ספורט מענפים שונים: כדורגל, כדורסל, טניס ועוד.
אתרים כגון אתר הווינר מעלים את המוטיבציה להשתמש בתוכנת מחשב לניבוי תוצאות משחקי ספורט, ובכך "לפצח" את מערכת ההימורים.
כדורגל הוא ענף הספורט הקבוצתי הפופולרי והנפוץ ביותר בעולם. ענף ההימורים בו מגלגל כ 25 מיליארד דולר בשנה.
במשחק הכדורגל משתתפות שתי קבוצות, שמטרת כל אחת מהן היא הכנסת כדור המשחק לתוך שערה של השנייה - "הבקעת גול". המנצחת בתום המשחק היא הקבוצה שהבקיעה את מירב הגולים בתום הזמן החוקי של המשחק.

ישנם המון טכניקות לחיזוי תוצאות משחק כדורגל, הבסיסית תהיה להשוות את מספר הגולים המצטבר/הממוצע של הקבוצות המתחרות.
אך, במציאות, תוצאות המשחק מושפעות דרסטית מתכונות רנדומליות נוספות: כושר השחקנים, שחקני מפתח נעדרים, מורל הקבוצה, עידוד הקהל, וכו'. תכונות אלו מובילות לאי תאימות בין ביצועי הקבוצה במשחק הנוכחי לעומת משחקי עבר.
מטרת פרויקט זה היא להשוות טכניקות שונות של למידת מכונה על מנת לקבל את אחוז הדיוק המירבי של חיזוי תוצאות המשחק.
הפרויקט משתמש בטכניקות למידה מונחית - Supervised - אלגוריתמים בענף למידת מכונה שבהם לומדים על בסיס אוסף דוגמאות פתורות.
בהינתן נתוני הקבוצות היריבות, נרצה לחזות את תוצאת המשחק מתוך התוצאות האפשריות: הקבוצה המארחת תנצח - ניצחון / תיקו / הקבוצה המתארחת תנצח - הפסד.

חלק ראשון - ארגון המידע

האתגר הראשוני והעיקרי הינו למצוא מאגר מידע איכותי של משחקי עבר, לסדר אותו ולבחור את התכונות שבהם נשתמש.
בפרויקט זה נשתמש במאגר המידע 'Kaggle European Soccer Database' שאסף מידע ממספר מקורות על משחקי עבר בליגות העיקריות באירופה בשנים 2008-2016.
בפרויקט זה התמקדנו במידע על הליגה האנגלית.
מאגר המידע הנ"ל מאורגן בטבלאות הבאות:

	type	name	tbl_name	rootpage
0	table	sqlite_sequence	sqlite_sequence	4
1	table	Player_Attributes	Player_Attributes	11
2	table	Player	Player	14
3	table	Match	Match	18
4	table	League	League	24
5	table	Country	Country	26
6	table	Team	Team	29
7	table	Team_Attributes	Team_Attributes	2

לכן ראשית, חיברנו את הטבלאות העיקריות - league, match, team, country לטבלה אחת המכילה את כל המידע הרלוונטי על הליגה האנגלית שתשמש בתור מאגר המידע הבסיסי.

	id	country_name	league_name	season	stage	date	home_team	away_team	home_team_goal	away_team_goal
0	1730	England	England Premier League	2008/2009	1	2008-08-16 00:00:00	Arsenal	West Bromwich Albion	1	0
1	1731	England	England Premier League	2008/2009	1	2008-08-16 00:00:00	Sunderland	Liverpool	0	1
2	1732	England	England Premier League	2008/2009	1	2008-08-16 00:00:00	West Ham United	Wigan Athletic	2	1
3	1734	England	England Premier League	2008/2009	1	2008-08-16 00:00:00	Everton	Blackburn Rovers	2	3
4	1735	England	England Premier League	2008/2009	1	2008-08-16 00:00:00	Middlesbrough	Tottenham Hotspur	2	1
...	...	...	...	...	...	...	...	...	...	...
3035	4705	England	England Premier League	2015/2016	38	2016-05-15 00:00:00	Stoke City	West Ham United	2	1
3036	4706	England	England Premier League	2015/2016	38	2016-05-15 00:00:00	Swansea City	Manchester City	1	1
3037	4707	England	England Premier League	2015/2016	38	2016-05-15 00:00:00	Watford	Sunderland	2	2
3038	4708	England	England Premier League	2015/2016	38	2016-05-15 00:00:00	West Bromwich Albion	Liverpool	1	1
3039	4702	England	England Premier League	2015/2016	38	2016-05-17 00:00:00	Manchester United	Bournemouth	3	1

3040 rows × 10 columns

בשלב הבא, נרצה להוסיף עוד תכונות וסטטיסטיקות רלוונטיות שעליהם נאמן את המודלים.
כמובן שהעמודה הראשונה שיש להוסיף הינה תוצאת המשחק - הערך שאותו נחזה.
תוצאת המשחק- H - הקבוצה המארחת ניצחה ("ניצחון")
D- תיקו
A - הקבוצה המתארחת ניצחה ("הפסד")
לאחר התייעצות עם מבינים בתחום, הוספנו בעזרת חישובים ידניים את התכונות הבאות אשר יכולות להשפיע על תוצאות המשחק:

- ממוצע הגולים של כל קבוצה עד למשחק הנוכחי
- התוצאה הנפוצה במשחקים שבהם הקבוצות שיחקו זו נגד זו

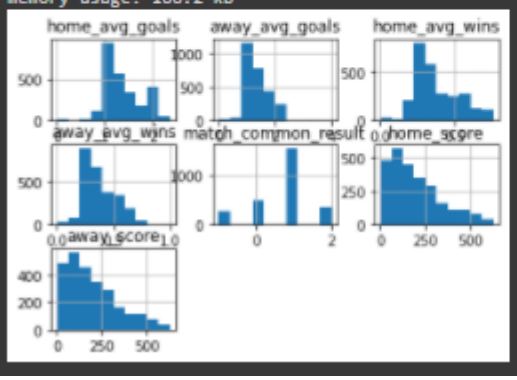
- סיכוי כל קבוצה לנצח (אחוז הניצחונות שלה מכלל המשחקים עד למשחק הנוכחי)
- ניקוד משוקלל של הקבוצה עד כה (0 נק' על כל הפסד, נק' 1 על כל תיקו, ו 3 נק' על כל ניצחון)

	id	country_name	league_name	season	stage	date	home_team	away_team	home_team_goal	away_team_goal	result	home_avg_goals	away_avg_goals	home_avg_wins	away_avg_wins	match_common_result	home_score	away_score
0	1730	England	England Premier League	2008/2009	1	2008-08-16 00:00:00	Arsenal	West Bromwich Albion	1	0	H	0.000000	0.000000	0.000000	0.000000	-1	0	0
1	1731	England	England Premier League	2008/2009	1	2008-08-16 00:00:00	Sunderland	Liverpool	0	1	A	0.000000	0.000000	0.000000	0.000000	-1	0	0
2	1732	England	England Premier League	2008/2009	1	2008-08-16 00:00:00	West Ham United	Wigan Athletic	2	1	H	0.000000	0.000000	0.000000	0.000000	-1	0	0
3	1734	England	England Premier League	2008/2009	1	2008-08-16 00:00:00	Everton	Blackburn Rovers	2	3	A	0.000000	0.000000	0.000000	0.000000	-1	0	0
4	1735	England	England Premier League	2008/2009	1	2008-08-16 00:00:00	Middlesbrough	Tottenham Hotspur	2	1	H	0.000000	0.000000	0.000000	0.000000	-1	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3035	4705	England	England Premier League	2015/2016	38	2016-05-15 00:00:00	Stoke City	West Ham United	2	1	H	1.056106	1.226415	0.320132	0.301887	0	377	314
3036	4706	England	England Premier League	2015/2016	38	2016-05-15 00:00:00	Swansea City	Manchester City	1	1	D	1.227513	1.996700	0.328042	0.577558	1	237	585
3037	4707	England	England Premier League	2015/2016	38	2016-05-15 00:00:00	Watford	Sunderland	2	2	D	1.027027	1.092409	0.324324	0.257426	2	44	325
3038	4708	England	England Premier League	2015/2016	38	2016-05-15 00:00:00	West Bromwich Albion	Liverpool	1	1	D	1.147170	1.749175	0.283019	0.495050	1	297	525
3039	4702	England	England Premier League	2015/2016	38	2016-05-17 00:00:00	Manchester United	Bournemouth	3	1	H	1.910891	1.189189	0.630363	0.297297	1	630	42

3040 rows × 18 columns

לסיום, חילקנו את המידע לתכונות וערך חזוי, נרמלנו את המידע, ומחקנו עמודות שלא רלוונטיות לאימון מודל (id, data, season וכו') + את כל המשחקים מעונה 2008/2009 (מאחר ואין על עונה זו מספיק נתונים).

```
Data columns (total 7 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   home_avg_goals       2660 non-null   float64
 1   away_avg_goals       2660 non-null   float64
 2   home_avg_wins        2660 non-null   float64
 3   away_avg_wins        2660 non-null   float64
 4   match_common_result  2660 non-null   int64
 5   home_score           2660 non-null   int64
 6   away_score           2660 non-null   int64
dtypes: float64(4), int64(3)
memory usage: 166.2 KB
```



המידע הנותר כולל רק נתונים מספריים רלוונטיים באותו סדר גודל.

קיבלנו: מספר התכונות: n= 7

מספר הדוגמאות: m = 2660

חלק שני- אימון המודלים השונים לסיווג תוצאת המשחק

בהינתן התכונות והערך החזוי, נרצה למצוא את המודל הטוב ביותר.

לשם השוואת המודלים נחלק את הדאטה ל-3 חלקים:

- Train – 60%

- Validation – 30%
- Test – 10%

התהליך יראה כך:

1. לכל מודל - נלמד על ה train
2. נחפש מי מבין המודלים+הפרמטרים לכל מודל נותן תוצאה טובה ביותר על ה- validation ונבחר מודל זה.
3. נלמד את המודל הכי טוב על train + validation
4. נסכם מהי איכות המודל לפי התוצאה על קבוצת ה- test

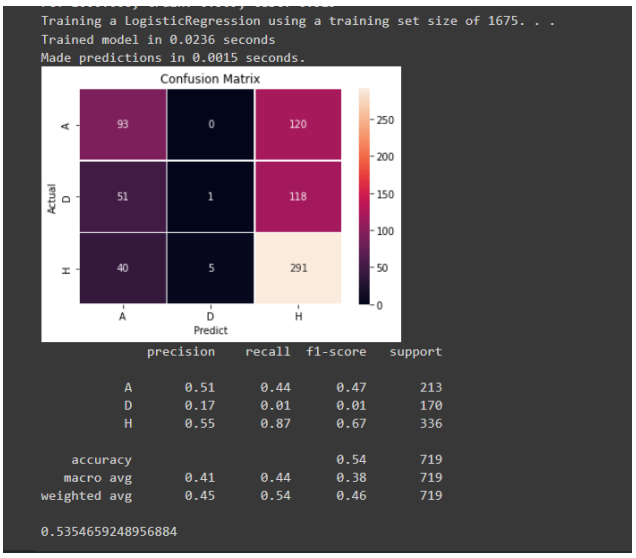
המודל הראשון - Logistic regression

$$y = \beta_0 + \beta_1 x_1 + ... + \beta_m x_m + \epsilon$$

מודל זה מבוסס על "רגרסיה לינארית" שבו בונים קו ישר במרחב שמפריד בין קבוצות בינאריות 0/1.
ב רגרסיה לוגיסטית, המודל מחשב את הסיכוי של השייכות לכל מחלקה, וכשמגיע קלט חדש, משייכים אותו למחלקה עם ההסתברות הגבוהה ביותר.
בכוונון הפרמטרים של המודל נתמקד ב אופטימיזציה של הפרמטר המשמעותי c - פרמטר זה קובע את מרחב החיפוש של המקדמים - עד כמה המודל יהיה מודל מסובך.
(פרמטר זה מוסיף את האילוץ שנורמת המשקלים תהיה שווה ל c - נשתמש בנורמה הדיפולטיבית - l2 .)
על מנת לבדוק האם יש overfitting של המודל, הדפסנו בכל איטרציה את הדיוק המתקבל לאחר בדיקה על ה train ועל ה test.
הc שנתן לנו את התוצאה הטובה ביותר היה 0.1.
כפי שניתן לראות עבור כל המשתנים אחוז הדיוק בין האימון לבדיקה הוא מאוד קרוב, מה שאומר שלא מתקיים מצב של overfitting.

```
>c: 0.000, train: 0.454, test: 0.467
>c: 0.001, train: 0.497, test: 0.517
>c: 0.010, train: 0.515, test: 0.535
>c: 0.100, train: 0.509, test: 0.529
>c: 1.000, train: 0.510, test: 0.529
>c: 10.000, train: 0.509, test: 0.529
>c: 100.000, train: 0.509, test: 0.529
>c: 1000.000, train: 0.509, test: 0.529
```

לאחר מכן אימנו את המודל על ה test , ובדקנו את רמת הדיוק על ה validation .



ניתן לראות שה accuracy -אחוז הדיוק הכללי של המודל הינו 53.5%.
אך אם מסתכלים היטב על המודל אחוזי הדיוק לכל מחלקה שונים מאוד!
מתוך סך המשחקים שבהם קבוצת הבית ניצחה המודל חזה נכון 86% לעומת רק 1% מתוך המשחקים שהסתיימו בתיקו.
(יש לשים לב שבכל הרצה האחוזים משתנים מעט בשל הרנדומליות בחלוקת הדאטה.)
התוצאות מתאימות למציאות ההימורים, כאשר סטטיסטית לקבוצת הבית יותר סיכוי לנצח, ותפיסת "תיקו" נחשבת יותר קשה לחיזוי.

המודל השני - AdaBoost

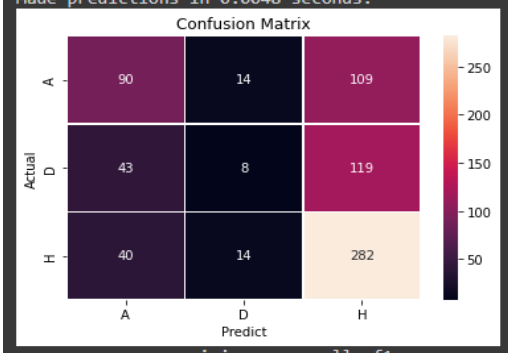
Adaboost הוא אלגוריתם למידה המשכלל אלגוריתמים חלשים רבים לאלגוריתם חזק.
זהו אלגוריתם איטרטיבי שבכל שלב לומד מהטעויות ומשתפר.
אנו נשתמש ב adaboost מבוסס על עצי החלטה ו בעזרת כוונון הפרמטרים נחפש את המודל שיתן לנו את אחוז הדיוק הטוב ביותר.
הפרמטרים החשובים ביותר הם:

base_estimator:
האלגוריתם הבסיסי שאותו נשפר. לא נשנה את האלגוריתם הדיפולטיבי - עצי החלטה-
אך נכוונן את עומק העץ המקסימלי מתוך הערכים [4,10,50,100].
ככל שהעץ עמוק יותר כך המודל יהיה מורכב יותר ו עלול להיווצר מצב של overfitting.

n_estimators:
מספר המודלים לאמן - במקרה שלנו מספר העצים מתוך הערכים [10,50,100,500]

learning_rate:
משקל כל מודל שאימנו למודל הסופי, הדיפולטיבי הינו 1. נבחר מתוך הערכים (0.1-1.2 בקפיצות של 0.1)
עבור כל שילוב מהפרמטרים ניצור מודל, נאמן אותו על train ונבחן אותו על validation.

```
Best Adaboost parameters and score:
n: 10 d: 4 r: 0.200 0.5257301808066759
Training a AdaBoostClassifier using a training set size of 1675. . .
Trained model in 0.0855 seconds
Made predictions in 0.0048 seconds.
```



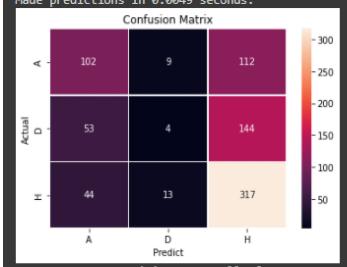

```
              precision    recall  f1-score   support

           A       0.52      0.42      0.47       213
           D       0.22      0.05      0.08       170
           H       0.55      0.84      0.67       336

    accuracy                           0.53       719
   macro avg       0.43      0.44      0.40       719
weighted avg       0.47      0.53      0.47       719

0.5285118219749653
```

כפי שניתן לראות הפרמטרים שנותנים לנו את הדיוק הטוב ביותר הם:
מספר העצים = 10, עומק העץ = 4, קצב למידה = 0.2
ניתן לראות שה accuracy -אחוז הדיוק הכללי של המודל הינו 52.8% - מעט פחות טוב מרגרסיה לוגיסטית (ב 0.004%).
אך במדדים אחרים מודל זה יותר טוב - בחיזוי של תוצאת "תיקו" או "קבוצת החוץ ניצחה"..
הפרמטרים שקיבלנו שנותנים את הדיוק הטוב ביותר מאוד פשוטים, מה שמראה שכנראה יש שונות מאוד גדולה בדוגמאות.

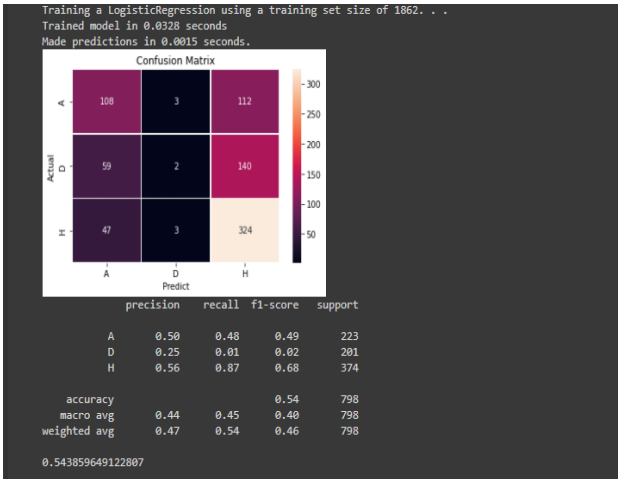
חלק שלישי- סיכום איכות המודלים

בחלק הקודם אחוזי הדיוק של המודלים שהשווינו יצאו מאוד קרובים. על כן, לא נבחר רק אחד מהם אלה נאמן את שני המודלים עם הפרמטרים הטובים ביותר שמצאנו על כל קבוצת ה train ו נשווה את התוצאות על הtest.

```
Training a AdaBoostClassifier using a training set size of 1862. . .
Trained model in 0.1008 seconds
Made predictions in 0.0049 seconds.
```




```
              precision    recall  f1-score   support

           A       0.51      0.46      0.48       223
           D       0.15      0.02      0.04       201
           H       0.55      0.85      0.67       374

    accuracy                           0.53       798
   macro avg       0.41      0.44      0.40       798
weighted avg       0.44      0.53      0.46       798

0.5300751879699248
```



ואלו המסקנות מההשוואה:

1. כצפוי שני המודלים השיגו אחוז דיוק גבוה יותר כאשר ניתן להם יותר דוגמאות - Logistic Regression - 0.543, AdaBoost - 0.53.
2. ההבדל בין המודלים גדל (0.013%), ו אפשר להסיק מכך שעבור המידע והתכונות שלנו logisticRegression מהווה מודל יותר טוב.

מסקנה זו מגיעה בהפתעה ובניגוד להשערותינו הראשונית ש adaboost יהווה מודל יותר מדויק - כל מטרתו הינה להגיע לאחוז דיוק גבוה יותר על סמך שקלול של המון מודלים פשוטים ולמידה מטעויות.

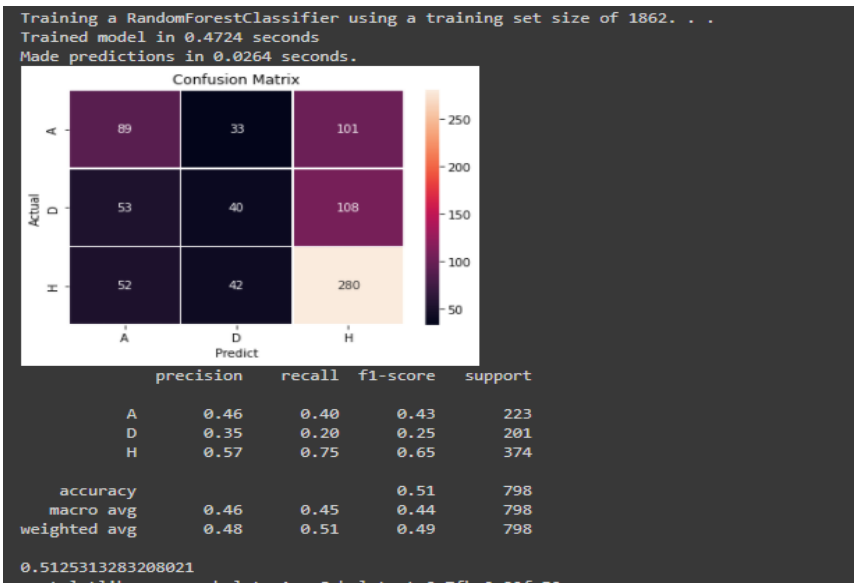
אנחנו חושבים שקיבלנו תוצאה זאת מאחר שהמידע מתפלג הסתברותית (ניתן לראות זאת בהתפלגות של התכונות) ו logisticRegression הינו מודל מבוסס על הסתברות.

חלק רביעי- דירוג התכונות שהוספנו

כ סיכום של הפרויקט רצינו להשתמש במודל נוסף וכמו כן לדרג את התכונות שחישבנו ידנית.

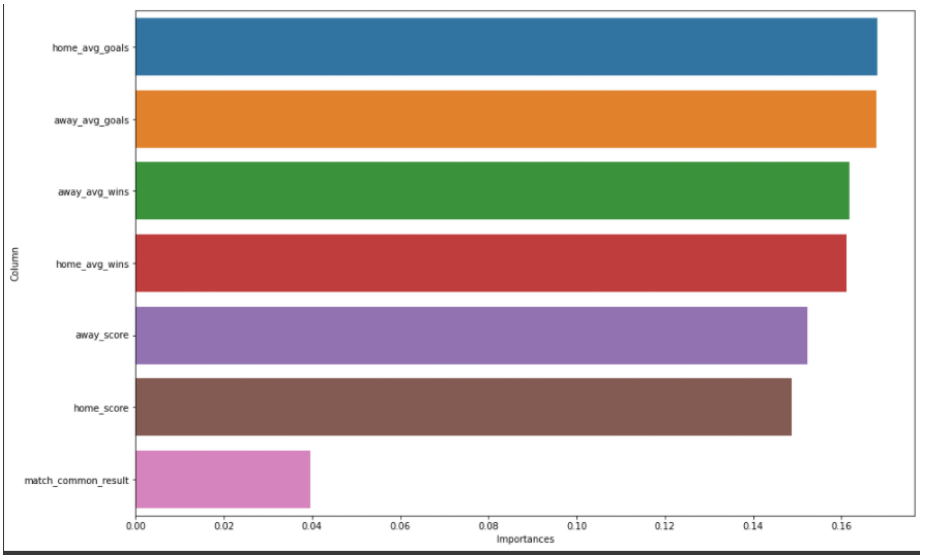
המודל השלישי - Random Forest

אלגוריתם זה בנוי מאוסף של עצי החלטה רנדומלים הבנויים כך שכל צומת בנוי על סמך תכונה אחרת. לכן החלטנו להשתמש באלגוריתם זה על מנת לדרג את חשיבות התכונות.



נשים לב כי בעת הרצת האלגוריתם עם ערכים דיפולטיבים קיבלנו את אחוז הדיוק הנמוך ביותר עד כה- 0.512%.

דירוג התכונות שהתקבל:



ניתן לראות כי כל התכונות שחשבנו כי משפיעות אכן חשובות מלבד match_common_result ועל כן לא ננסה לחתוך תכונות ולאמן את המודלים רק על חלקם.

מסקנות

למרות עבודתינו הקשה ליצור תכונות נוספות שבהגיון הפשוט משפיעות על תוצאות המשחק, משחקי הכדורגל מושפעים מהמון תכונות רנדומליות נוספות שלא ניתן לכמת או לחזות לפני המשחק (שחקן מפתח נפצע במהלך המשחק, מזג האוויר, עידוד הקהל, כושר הקבוצה וכו')

אנו מאמינות שזו הסיבה העיקרית שלא הצלחנו להביא לתוצאה משמעותית כדי להחליט את תוצאות המשחק (0.5% - שווה ערך להטלת מטבע) ..
אך הרנדומליות בתוצאה היא המובילה לאהבה של המשחקים, ריגוש בהימורים ו אדרנלין בעת צפיה במשחק הנובע מהתוצאה הלא חזויה.
אפשר להרחיב פרויקט זה עבור מידע עם עוד תכונות - למשל לבסס פרופיל של הקבוצה על בסיס השחקנים, המאמן. להוסיף עוד דוגמאות מעוד ארצות. לנסות מודלים נוספים וכו'.
אחוז דיוק משמעותי של המודלים יכול להוביל לרווח רב, לכן חיזוי משחקי ספורט הפך לענף בפני עצמו בתוך תחום הלמידה חישובית. בפרויקט זה יצא לנו לטעום רק מעט מ מורכבותו.

מקורות:

Predicting Football Results Using Machine Learning Techniques

European Soccer Database

Machine Learning Algorithms for Football Prediction using statistics from Brazilian championship data

Predicting the Winning Team with Machine Learning

https://www.football-data.co.uk/data.php

guide-to-football-and-soccer-data-and-apis/

Data Analysis using SQL

Match Outcome Prediction in Football

Online Sports Betting Market Size, Share, Forecast 2027 | MRFR

How to Develop an AdaBoost Ensemble in Python