[slide 1]

# Presentatie project ai

## Inleiding

[slide 2]

Dag meneer, welkom bij onze presentatie van het project.

[slide 3]

We gaan het tijdens deze presentatie hebben over:

* Een beschrijving van de wijzigingen aan de originele dataset,
* De behaalde resultaten voor de classificatie en conclusies,
* Moeilijkheden die we hebben ondervonden in de loop van het project.

[slide 4-5]

## Beschrijving wijzigingen aan originele dataset

Om het project te maken hebben we twee tabellen nodig. De “**df\_match**” en “**df\_player\_attributes**”.

Match bevat alle data van een **match**, bv **league id**, **land id**, **seizoen**, **stage**, **goals**, **team api** **ids**, en **allerlij numerieke** **data** van de match

Player attributes bevat info over een speler zoals bv **links** of **rechtsvoetig**, **rating**, en **allerlij numerieke data** van een speler.

Eerst hebben we wat aan data exploratie gedaan en vervolgens zijn we begonnen aan het opkuisen/bewerken van de data.

[slide 6]

We hebben alle kolommen die de gokkansen bijhouden gedropt. Dit zijn alle kolommen vanaf “**B365H**”. Ook hebben we de kolommen gedropt die informatie bijhouden over het verloop van de wedstrijd zoals **goal**, **shoton**, **shotoff**, **foulcommit**, **card**, **cross**, **corner**, **possession**. Ook de kolommen **id**, **country**\_id, **league**\_id, **season**, **stage**, **date**, **match\_api\_id**, **home\_team\_api\_id**, **date** en **away\_team\_api\_id** hebben we gedropt.

[slide 7]

Vervolgens hebben we alle rijen gedropt waar **minstens één** **speler id** niet gekend is. Dan houden we nog 13 rijen over met NaN waarden en om het simpel te houden en complicaties te voorkomen in de volgende stappen hebben we deze allemaal gedropt (in plaats van elke rij te droppen waar 5 of meer NaN waarden in voorkomen).

[slide 8]

Hierna hebben we een **kolom toegevoegd** die weergeeft of de **thuisploeg gewonnen** heeft. 1 is thuisploeg win, 0 is gelijkspel en -1 is uit ploeg win.

Momenteel zijn we klaar in “df\_match”. We gaan nu over naar de “df\_player\_attributes”.

[slide 9]

De eerste stap die we hier hebben uitgevoerd is ervoor zorgen dat we de **meest recente** rij nemen van de speler in kwestie moest die speler meerdere keren voorkomen in de tabel.

Dan hebben wede kolommen **id**, **player\_fifa\_api\_id** en **date** gedropt.

Vervolgens hebben we een **one hot encoding** uitgevoerd op de **preferred foot** kolom en de **attacking\_work\_rate** en **defensive\_work\_rate** kolommen gedropt.

[slide 10]

Daarna hebben we voor elke match de som berekend van de attributen van de spelers van beide teams. We hebben dan 75 kolommen, twee keer 37 kolommen met de stats van het team en een kolom welke ploeg er gewonnen heeft.

En om af te sluiten hebben we de kolommen verplaatst, dat de eigenschappen van thuisploeg eerst komen en dan pas die van de uit ploeg.

[slide 11]

Om af te sluiten hebben we nog een standard scaler uitgevoerd op alle kolommen behalve de kolom die bijhoudt welke ploeg er wint en de dataset gesplit in train en test.

Dit waren alle stappen die we hebben uitgevoerd op de datasets qua opkuisen en voorbereiden voor de machine learning technieken toe te passen.

[slide 12]

## Resultaten bespreken

We hebben vier verschillende technieken gebruikt, namelijk: Random Forest, SVC, Logistic Regression en kNN.

[slide 13]

Bij Random Forest hebben we een GridSearch gedaan met criterion = entropy of gini, max\_depth = range van 2 tot 14 stapjes van 2, min\_samples\_split = 2, 10 of 50 en min\_samples\_leaf = 1, 10, 50, 100, of 500.

[slide 14]

Bij SVC hebben we een GridSearch gedaan met C = 0.001, 0.01, 0.1, 1, 10, 100 of 1000, gamma = 1, 0.1, 0.01, 0.001 of 0.0001 en kernel = rbf, poly of sigmoid.

[slide 15]

Bij Logistic Regression hebben we ook een GridSearch gedaan met C = 0.001, 0.01, 0.1, 1, 10, of 100, solver = netwon-cg, lbfgs, liblinear of sag, max\_iter = 2147483647 en tol = 0.0001, 0.001, 0.01, 0.1.

[slide 16]

Bij kNN hebben we ook een GridSearch gedaan met n\_neighbors = een range van 1 tot 30, algorithm = auto, ball\_tree, kd\_tree, brute en weights = uniform, distance.

[slide 17]

Als we gaan kijken naar de resultaten kunnen we concluderen dat Logistic Regression echt wel een monopolie heeft op de beste resultaten. Meer als 50 modellen met de beste resultaten zijn allemaal van het type Logistic Regression.

Het beste model dat wij getraint hebben is het model nummer 59. Dit model heeft een score van 0.5358479004460427 en een fit time van 0.4764587879180908 sec.

De hyperparameters van het model zijn:

C = 1, max\_iter = 2147483647, solver = liblinear, tol = 0.01.

[slide 18]

Verder hebben we ook nog wat tabellen en figuren gemaakt om de resultaten beter te begrijpen.

De SVC-techniek heeft een gemiddelde fit tijd van 20.48 sec, mediaan van 12.98 sec, minimum van 6.12 sec, en een maximum van 61.80 sec. Hij heeft een gemiddelde score van 0.46, mediaan van 0.45, minimum van 0.37 en een maximum van 0.53.

De kNN-techniek heeft een gemiddelde fit tijd van 0.06 sec, mediaan van 0.04 sec, minimum van 0.005 sec en een maximum van 0.16 sec. Hij heeft een gemiddelde score van 0.48, mediaan van 0.49, een minimum van 0.49 en een maximum van 0.50.

De LOR-techniek heeft een gemiddelde fit tijd van 0.70 sec, mediaan van 0.41 sec, minimum van 0.06 sec en een maximum van 7.39 sec. Hij heeft een gemiddelde score van 053, mediaan van 0.53 minimum van 0.52 en een maximum van 0.53.

En als laatste, de RFC-techniek heeft een gemiddelde fit tijd van 1.84 sec, mediaan van 1.68 sec, minimum van 0.61 sec en een maximum van 6.41 sec. Hij heeft een gemiddelde score van 0.52, mediaan van 0.52, minimum van 0.50 en een maximum van 0.53.

Hieruit kunnen we concluderen dat als je een heel snel model wilt, je best de kNN techniek gebruikt. Maar als je een iets trager maar nogsteeds snel model wilt dat accurater is, gebruik je best de LOR.

De RFC techniek is dubbel zo traag als LOR maar behaalt ongeveer dezelfde resultaten.

De SVC techniek is een absolute no-go omdat hij gemiddeld 20 keer zo traag is als de LOR techniek en hij behaalt slechtere resultaten.

Met dat gezegd te hebben, hebben we onze resultaten besproken en kunnen we overgaan naar het volgende deel van onze presentatie.

[slide 19]

## Moeilijkheden

Er waren enkele stukken waar we het moeilijker mee hadden in het project.

[slide 20]

Het eerste waar we even op vast zaten was het berekenen van de som van de stats van de spelers van een team met de apply functie. Na even hulp te vragen bij andere klas genoten hebben ze het ons uitgelegd en hebben we het probleen kunnen oplossen.

[slide 21]

Daarna zaten we even vast bij de pairplot. Hij nam heel erg lang om te maken. Meerdere uren zelfs. We hebben dan minder kolommen moeten nemen en een sample van de data. We moesten keuzes maken tussen welke kolommen we wel namen en welke niet. We hebben dit opgelost door 16 kolommen te nemen die de hoogste corellatie hadden met het label. Hiervan hebben we dan een pairplot gemaakt met een sample van de data en niet alles.

[slide 22]

En op het einde hebben we nog even vastgezeten met de SVC techniek omdat die heel lang nam om te trainen en dus hierdoor hebben we wel wat tijd verloren om telkens eens iets aan te passen en te rerunnen enzovoort.

[slide 23]

En met dat gezegd te hebben sluiten we onze presentatie af, bedankt voor uw aandacht en zijn er nog vragen?