* DB aanmaken voor alle informatie, die opgevraagd wordt wanneer de Flask app wordt gerund
* Gridsearch met groupkfold = year. Elk jaar wordt apart beoordeeld als validatieset.  
    
  !! Year moet ik dan niet droppen in preprocessing, maar pas in de group\_split\_data()  
    
  Group K-Fold

An example would be when there is medical data collected from multiple patients, with multiple samples taken from each patient. And such data is likely to be dependent on the individual group. In our example, the patient id for each sample will be its group identifier.

In this case we would like to know if a model trained on a particular set of groups generalizes well to the unseen groups. To measure this, we need to ensure that all the samples in the validation fold come from groups that are not represented at all in the paired training fold.

* Model fitten op data van 1950 tot en met 2016 bijvoorbeeld en dan voorspellen voor 2017 🡪 2017 als aparte dataframe maken, die opgeslagen wordt in een sqlite DB.

To do:  
- DB createn met alle preprocessed data inlcusief left\_driver, right\_driver, year en circuit. Deze droppen bij de voorspelling, maar is nodig om een row op te vragen met de flask app

* Functie schrijven die het aantal gewonnen duels optelt per rijder, op slaat in een dictionary en zo de winnaar, 2e en 3e teruggeeft.
* Homepage met menu met buttons die redirecten naar andere pagina’s
* Pagina met dashboards, waar gebruiker input kan leveren en een seaborn plot wordt gereturned
* Predicted\_prob tonen bij de prediction results

Nieuwe features:

* Kwalificatie
* Resultaat voorgaande jaar op het circuit: done
* Duel van vorige race: done

Run proces:

* Nieuwe data downloaden in form van CSV en plaatsen in map: new\_data
* Run-retrain met nieuwe data. Let op: neem alles minus het huidige jaar als trainingsdata
* Fill-db met alle nieuwe data

# New prediction of upcoming race

Voor nieuwe voorspellingen is de volgende handmatige data nodig:

* Results.csv 🡪 voor alle rijders dummy data invullen.
* Driverstandings.csv  
    
  Onderaan 20 dummy rows toevoegen (aangezien er 20 rijders zijn)  
    
  !! Let op: niet overal dummies invullen. Enkel bij de resultaten.  
    
  Voor results bij de volgende features dummies invoeren:
* Grid
* Position
* positionText
* positionOrder
* points
* laps
* time
* miliseconds
* fastestLap
* rank
* fastestLapTime
* fastestLapSpeed
* statusId

!! index, resultId en raceId doornummeren

Voor driverstandings voor de volgende features dummies invoeren:

* points
* position
* positionText
* wins

!! driverstandingsId en raceId doornummeren