NFL PLAY PREDICTOR

By Mohammed Ansari & Samy Metadjer

Machine Learning course

Paris Diderot University

DATASET

- Source: https://www.kaggle.com/maxhorowitz/nflplaybyplay2009to2016
- 408 000 lines of data with more than 100 parameters describing each play
- CSV file

CLEANING

- Parsing:
  - Each line contains all the data we need to know (Xs and Ys) about a play.
  - We extracted the data and stored it into the appropriate data structures

```
/Users/samy/PycharmProjects/td7/venv/bin/python /Users/samy/git/nfl-play-predictor/predictor.py
How many corrupted lines ?
144472
How many post processed lines in dataset.csv ?
49460

Process finished with exit code 0
```

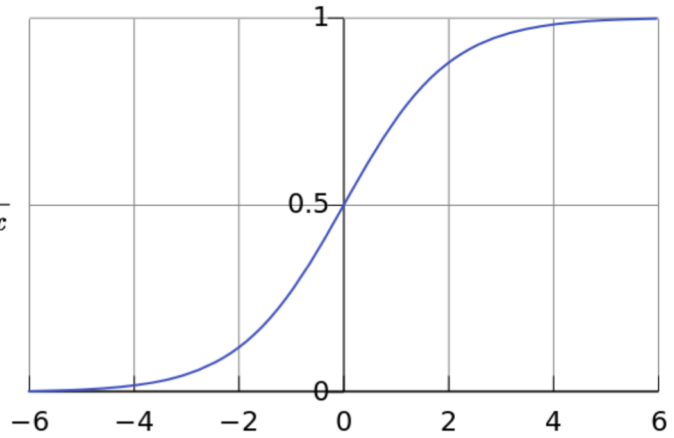
```
labelsXArray = ['Drive', 'down', 'TimeSecs', 'PlayTimeDiff', 'yrdline100', 'ydstogo', 'FirstDown', 'posteam',
                'PlayType', 'PassAttempt', 'RushAttempt', 'HomeTeam']

labelsYArray = ['Yards.Gained', 'Touchdown', 'Safety', 'InterceptionThrown', 'Reception']
```

ALGORITHMIC MODEL

- What kind of problem do we have ?
- What are the models solving the problem ?
- Which one did we choose ? Why ?

LA FONCTION LOGIT

$$f(x) = \frac{e^x}{1 + e^x}$$



Source : wikipedia.com

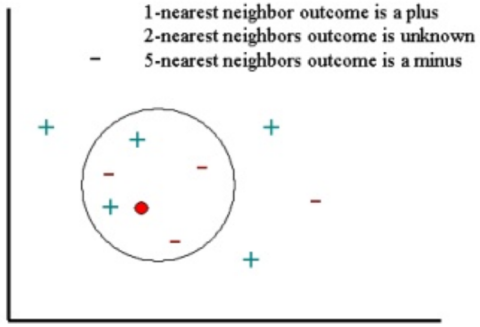
La régression logistique revient à appliquer cette fonction à une régression linéaire simple pour "pousser" les valeurs vers 0 et 1.

ALGORITHMIC MODEL

- What kind of problem do we have ?
- What are the models solving the problem ?
- Which one did we choose ? Why ?

LES K PLUS PROCHES VOISINS: PRINCIPE

K-plus proches voisins (K-nearest neighbors) : La valeur d'un point dépend de celles des points qui lui ressemblent.



Source : statsoft.com

ENCODING

- No need to encode with a specific method
- We mostly took the raw data, and encoded some specific parameters

```
if lineX[labelsX['PlayType']] != 'Pass' and lineX[labelsX['PlayType']] != 'Run':
    post_processed += 1
    continue
if lineX[labelsX['PlayType']] == 'Pass':
    lineX[labelsX['PlayType']] = float(-1)
if lineX[labelsX['PlayType']] == 'Run':
    lineX[labelsX['PlayType']] = float(1)
```

TUNING

- 102 parameters in the original dataset, it was way too much !
- We used crossed-validation for one model : K nearest neighbors
- We also used the weight provided by the LogisticRegression

TUNING

```
def find_best_k(train_x, train_y, dist_function, w):
    value = {}
    k_value = sampled_range(1, len(train_x) / 10, 20)

    kf = KFold(n_splits=10)
    for train_index, test_index in kf.split(train_x):
        X_train, X_test = numpy.array(train_x)[train_index], numpy.array(train_x)[test_index]
        y_train, y_test = numpy.array(train_y)[train_index], numpy.array(train_y)[test_index]
        for i in range(len(k_value)):
            if k_value[i] in value:
                value[k_value[i]] += eval_classifier(numpy.asfarray(X_train), y_train,
                                                     numpy.asfarray(X_test), y_test,
                                                     is_productive,
                                                     dist_function, w, k_value[i])
            else:
                value[k_value[i]] = eval_classifier(numpy.asfarray(X_train), y_train,
                                                    numpy.asfarray(X_test), y_test,
                                                    is_productive,
                                                    dist_function, w, k_value[i])
    best_v = 10000
    best_k = 0
    for k, v in value.items():
        if v < best_v:
            best_v = v
            best_k = k
    print 'Taux pour k =', best_k, ':', float(best_v) / float(10)
    return best_k
```

TRAIN / VALIDATION

- We split the dataset.csv into 2 files ‘test’ and ‘train’

RESULTS

```
/Users/samy/PycharmProjects/td7/venv/bin/python /Users/samy/git/nfl-play-predictor/predictor.py
How many corrupted lines ?
144472
How many post processed lines in dataset.csv ?
49460
0.554998194149 0.904338394794
(0.6222222222222222, 0.8571428571428571)

Process finished with exit code 0
```

DISCUSSION

- Nothing.