Kaggle Tips and Tricks

РАЗБОР ЗАДАЧИ CLASSIFY PRODUCTS INTO THE CORRECT CATEGORY

Сама суть

- Можно использовать результаты нескольких алгоритмов, а не одного
- В подавляющем большинстве случаев композиция алгоритмов даёт лучший результат, нежели какой-то один

Почему это работает

- Ошибки алгоритмов взаимно компенсируются
- На одних объектах (областях объектов) хорошо работают одни алгоритмы, на других – другие
- У разных алгоритмов может быть разная структура ответов

Смеси алгоритмов

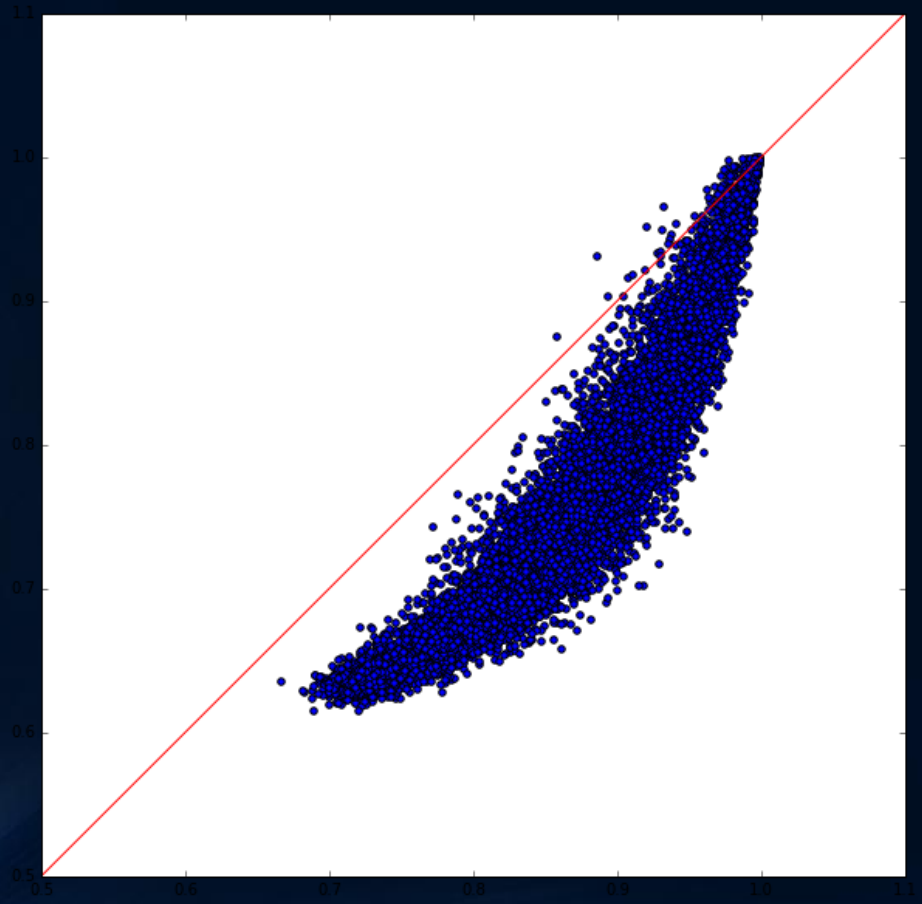
- Простейший вид композиции:
- $F = a_1 * F_1 + a_2 * F_2 + a_3 * F_3 \ldots$

- В смесях можно использовать разные алгоритмы, алгоритмы с разными параметрами, алгоритмы на разных признаках

Смеси алгоритмов

- Желательно использовать принципиально разные алгоритмы, наборы признаков
- Веса можно подбирать последовательно
- Веса могут быть отрицательными
- Не всегда два оптимальных алгоритма в смеси дадут оптимальный результат
- Чем больше параметров – тем больше шанс переобучиться

График алгоритм-алгоритм



Композиция алгоритмов

- $F = a_1 * F_1 + a_2 * f(F_2) + a_3 * g(F_3)$…

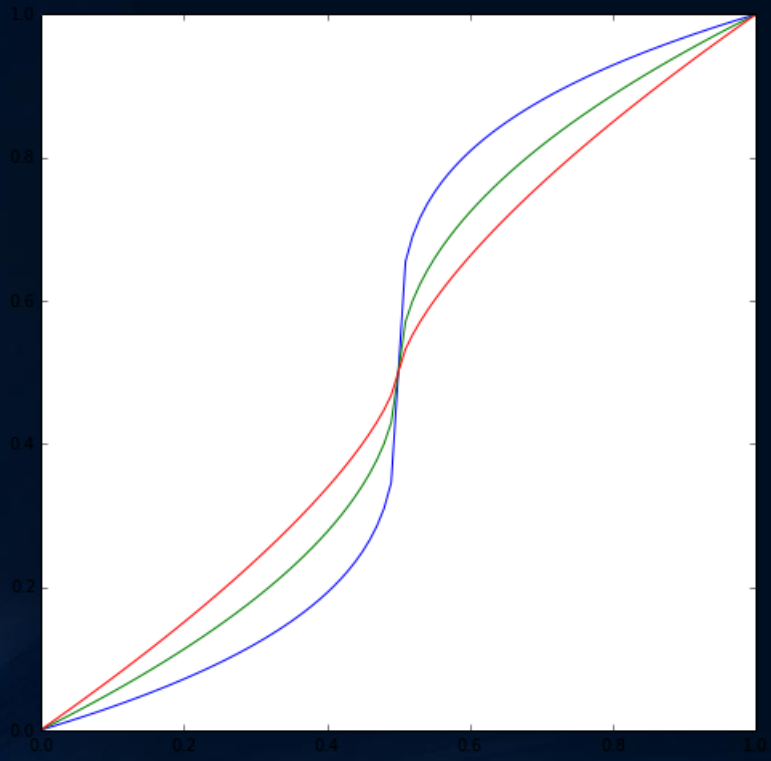
f, g – например степенные функции

- $F = f(a_1 * f^{-1}(F_1) + a_2 * f^{-1}(F_2) + a_3 * f^{-1}(F_3)$ …

f – например логистическая функция

Преобразование ответов

- new_y = 0.5 * ((2 * abs(y - 0.5)) ** beta) * sign(y - 0.5) + 0.5

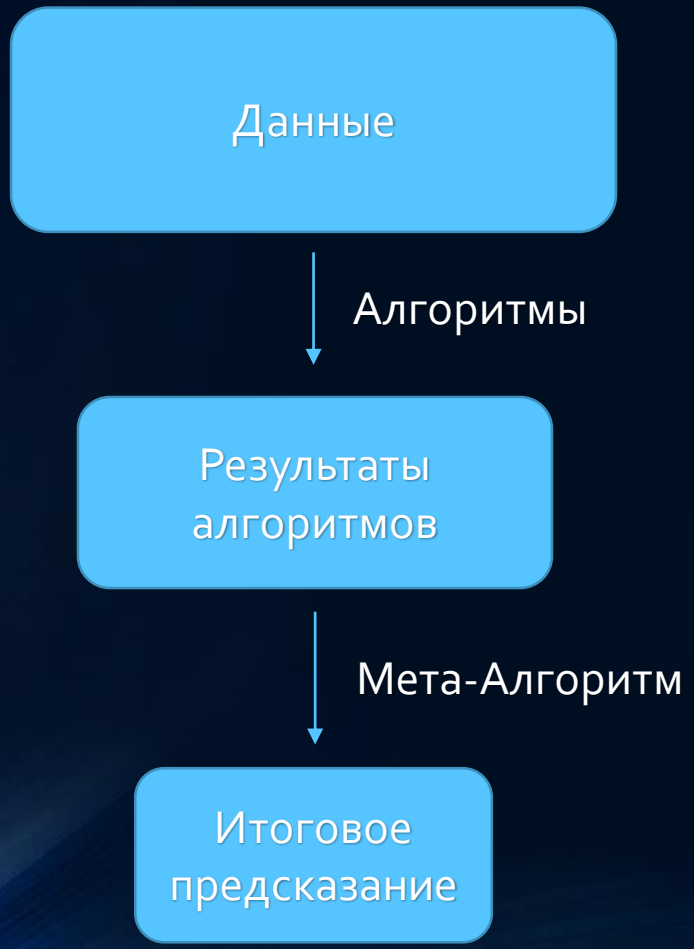


Стекинг алгоритмов

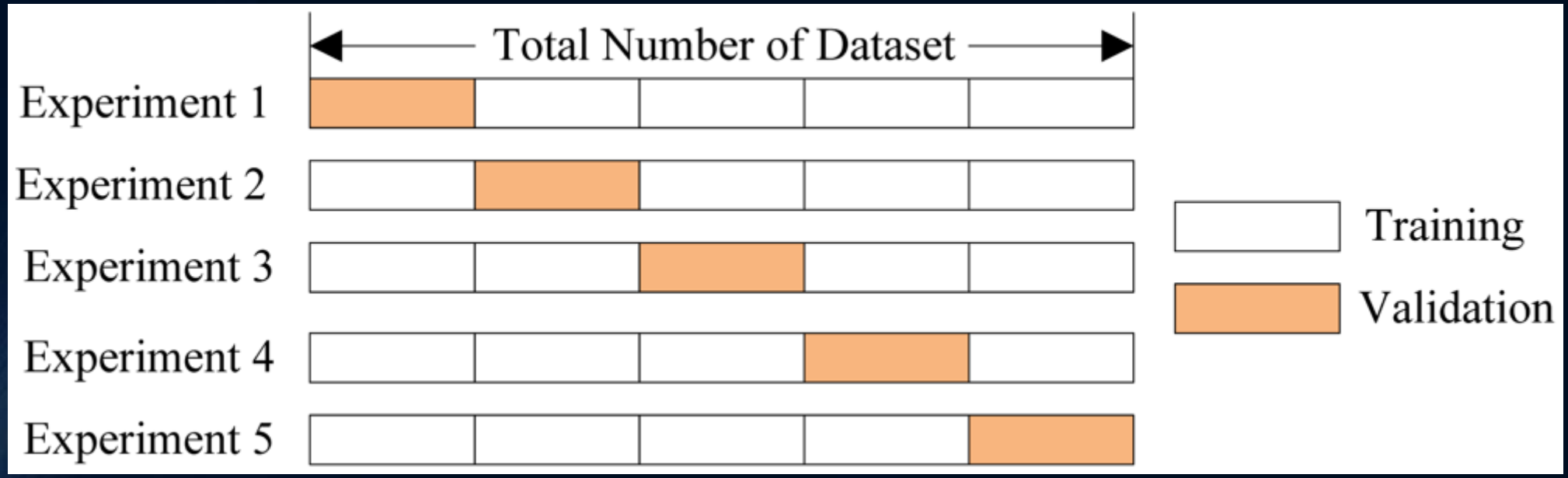
- Результаты работы алгоритмов можно использовать как полноценные признаки

Стекинг алгоритмов

Стекинг алгоритмов



K-Fold Стекинг



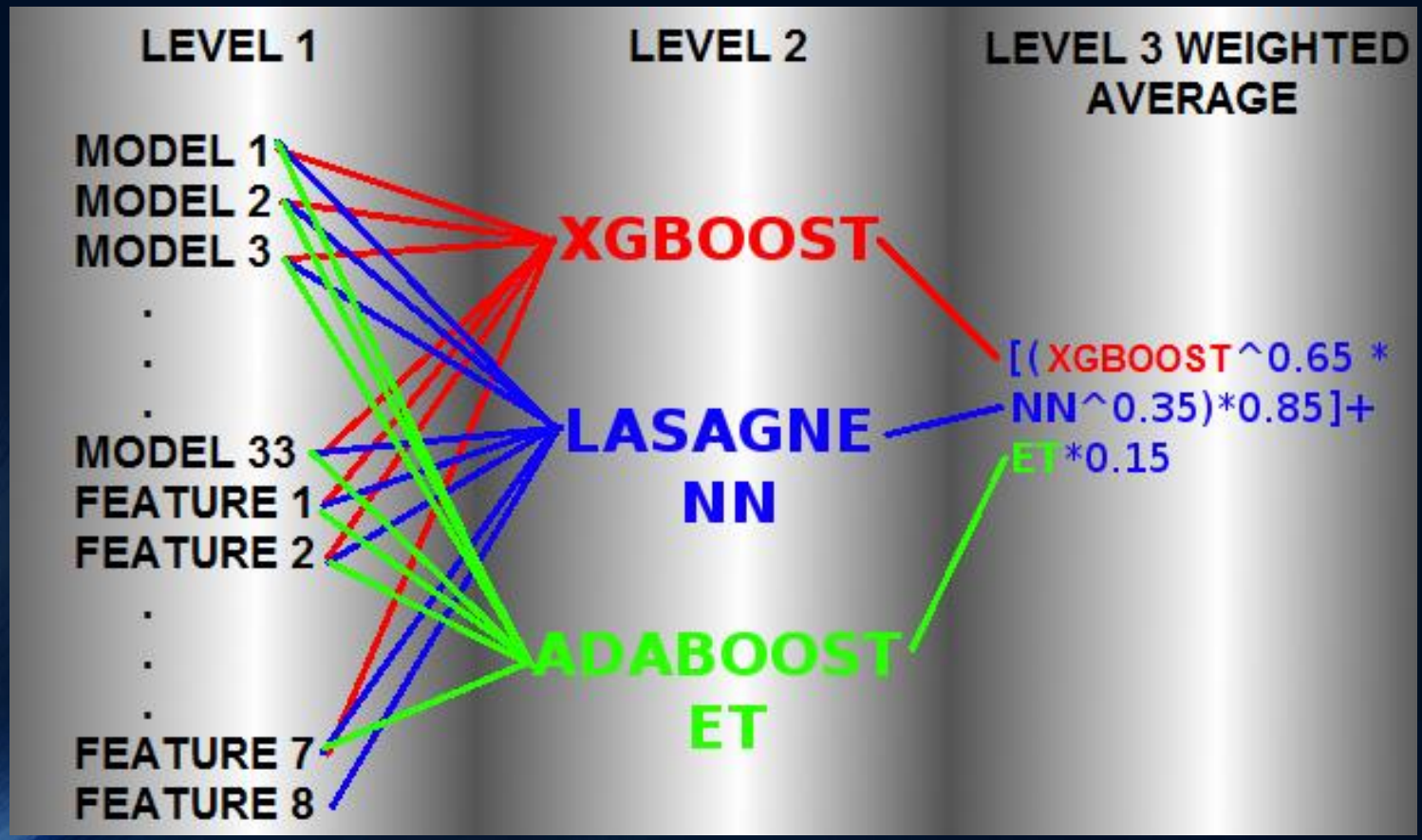
K-Fold Стекинг

- Первый вариант – предсказывать значения для теста по всему трейну.
- Нужно заново обучать модель.
- Предсказания для фолда и для теста имеют немного разные распределения.
- Используется больше объектов для обучения.

K-Fold Стекинг

- Второй вариант – предсказывать значения для теста по (k – 1) фолду и усреднять.
- Не нужно заново обучать модель.
- Распределения для предсказаний будут совпадать.
- Усредненные предсказания для дискретных методов могут преподнести неприятный сюрприз.

OTTO


LEVEL 1
LEVEL 2
LEVEL 3 WEIGHTED AVERAGE
MODEL 1
MODEL 2
MODEL 3
MODEL 33
FEATURE 1
FEATURE 2
FEATURE 7
FEATURE 8
XGBOOST
LASAGNE NN
ADABOOST ET
[(XGBOOST^0.65 * NN^0.35)*0.85]+ ET*0.15

Модель 1

- Язык:

Python

- Модель:

Logistic Regression (SciKit Learn)

- Признаки:

Log(X+1)

Модель 2

- Язык:

Python

- Модель:

Extra Trees Classifier (SciKit Learn)

- Признаки:

Log(X+1)

Модель 3

- Язык:

Python

- Модель:

KNeighborsClassifier (SciKit Learn)

- Признаки:

Log(X+1)

Модель 4

- Язык:

Python

- Модель:

KNeighborsClassifier (SciKit Learn)

- Признаки:

Scale(Log(X+1))

Модель 5, 6

- Язык:

Python

- Модель:

Bag of 2,6 NNs (Lasagne)

- Признаки:

Scale(Log(X+1))

Модель 7

- Язык:

R

- Модель:

Random Forest

- Признаки:

X

Модель 8

- Язык:

-

- Модель:

LibFM

- Признаки:

Sparse(X)

Factorization machines

- Линейная модель:

$$\phi(\mathbf{w}, \mathbf{x}) = \mathbf{w}^T \mathbf{x} = \sum_{j \in C_1} w_j x_j$$

- Факторизационные машины:

$$\phi(\mathbf{w}, \mathbf{x}) = \sum_{j_1, j_2 \in C_2} \langle \mathbf{w}_{j_1}, \mathbf{w}_{j_2} \rangle x_{j_1} x_{j_2}$$

Модель 9

- Язык:

R

- Модель:

SofiaML with learner_type="logreg-pegasos", loop_type="balanced-stochastic"

- Признаки:

Scale(X)

Модель 10

- Язык:

R

- Модель:

Xgboost One Agaist All

- Признаки:

X + sum(zeros) by row

Модель 11

- Язык:

R

- Модель:

Xgboost Multiclass

- Признаки:

Scale(X), 7 Kmeans features with different number of clusters: 9, 10, 11, 12, 13, 14 and 15, rowSums(X==0), rowSums(Scale(X)>0.5), rowSums(Scale(X)< -0.5)

Модель 12

- Язык:

R

- Модель:

Bag of 20 H2O NN

- Признаки:

Log(X+1)

Модель 13-15

- Язык:

R

- Модель:

3x Xgboost Multiclass

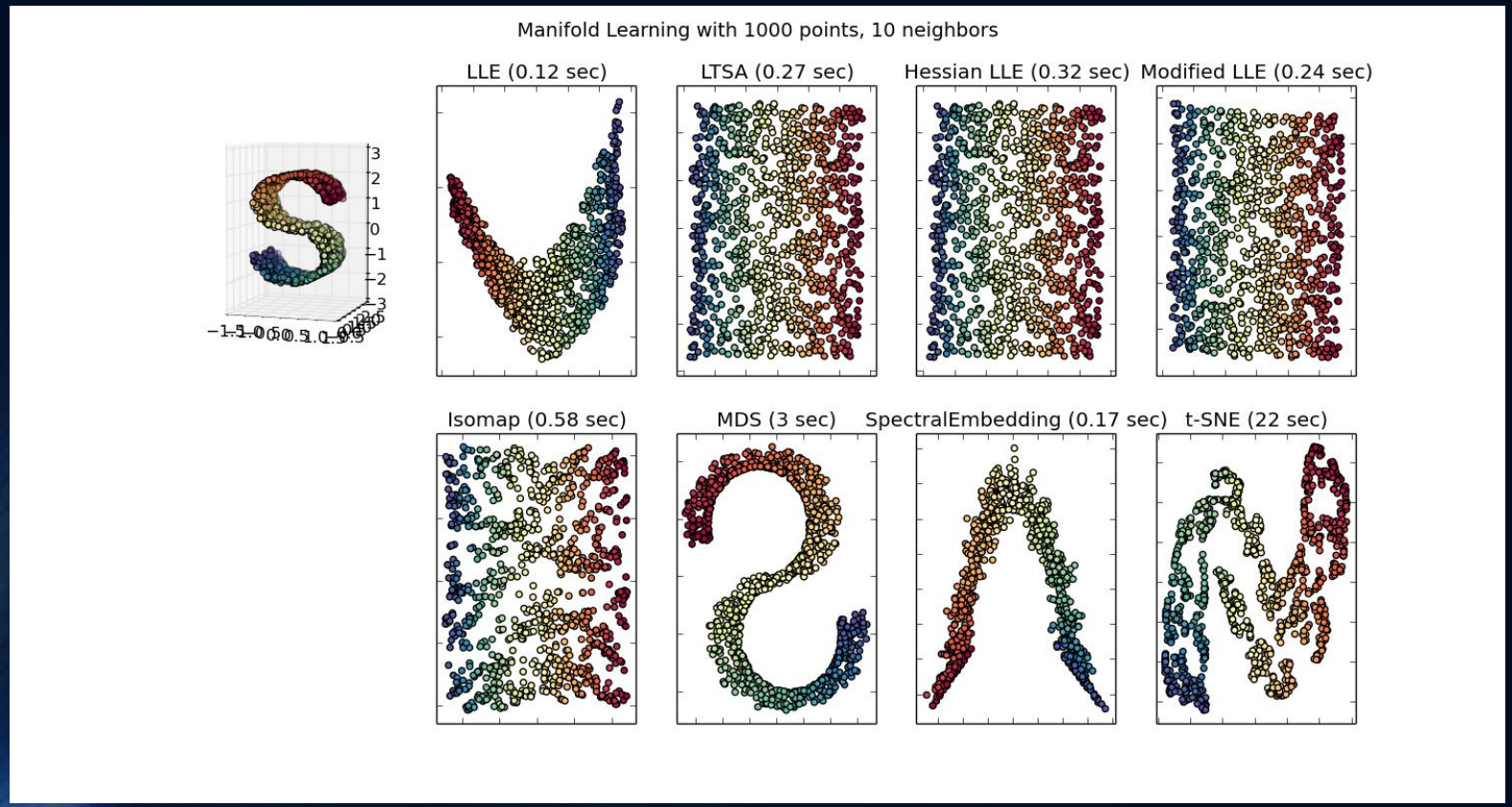
- Признаки:

X, rowSums(X==0), rowSums(Scale(X)>0.0), T-sne, Kmeans over T-sne with 1024 centers, Kmeans over log(X+1) with 1024 centers

T-sne

- t-distributed stochastic neighbor embedding
- Алгоритм машинного обучения для уменьшения размерности
- Без учителя
- Есть в Scikit-Learn, но лучше пользоваться внешней библиотекой

Manifold Learning



Модель 16, 17

- Язык:

R

- Модель:

SofiaML with learner_type="logreg-pegasos", loop_type="balanced-stochastic“, loop_type="combined-roc"

- Признаки:

Scale(X), T-sne, 3 feature iteration of 13 most importante features according Random Forest Importance

Модель 18

- Язык:

Python

- Модель:

XGboost, bagged 30 times

- Признаки:

X

Настройки XGboost

- 1. Большое число итераций, малый параметр бустинга
- 2. Малое число итераций, большой параметр бустинга + усреднение

Модель 19

- Язык:

Python

- Модель:

Lasagne 2-layer NN, bagged 30 times with different nets and number of epochs

- Признаки:

Scale(X), Scale(int(X==0)), Scale(Log(X+1))

Настройки NN

- layers = [('input', InputLayer),
  ('dropoutf', DropoutLayer),
  ('denseo', DenseLayer),
  ('dropout', DropoutLayer),
  ('dense1', DenseLayer),
  ('dropout2', DropoutLayer),
  ('output', DenseLayer)]

- net = NeuralNet(
  layers = layers,
  input_shape = (None, shape(Xtrain)[1]),
  dropoutf_p = 0.25,
  denseo_num_units = 1024,
  dropout_p = 0.25,
  dense1_num_units = 512,
  dropout2_p = 0.25,

Модель 20

- Язык:

Python

- Модель:

Lasagne 3-layer NN, bagged 30 times with different nets and number of epochs

- Признаки:

Scale(X), Scale(int(X==0)), Scale(Log(X+1))

Настройки NN

- layers = [('input', InputLayer),
  ('dropoutf', DropoutLayer),
  ('denseo', DenseLayer),
  ('dropout', DropoutLayer),
  ('dense1', DenseLayer),
  ('dropout2', DropoutLayer),
  ('dense2', DenseLayer),
  ('dropout3', DropoutLayer),
  ('output', DenseLayer)]
- net = NeuralNet(layers=layers,

  input_shape=(None, shape(Xtrain)[1]),
  dropoutf_p=0.05,
  denseo_num_units=1024,
  dropout_p=0.5,
  dense1_num_units=512,
  dropout2_p=0.5,
  dense2_num_units=256,
  dropout3_p=0.5,

NN

- Усреднять не только по разным запускам, но и по эпохам:

80 – 0.46

90 – 0.45

100 -0.46

110 – 0.48

120 – 0.49

- 80 + 90 + 100 – 0.44

Модель 21

- Язык:

Python

- Модель:

KNN with standard metric, 4 neighbours

- Признаки:

X, int(X==0)

Модель 22

- Язык:

Python

- Модель:

KNN with standard metric, 4 neighbours

- Признаки:

X, int(X==0), Log(X+1)

Модель 23-32

- Язык:

Python

- Модель:

KNN with BrayCurtis metric, neighbours: 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024

- Признаки:

X

Модель 33-37

- Язык:

Python

- Модель:

KNN with BrayCurtis metric, neighbours: 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024

- Признаки:

T-Sne

Признак 1

- Язык:

Python

- Модель:

Distances to nearest neighbours of each classes

- Признаки:

X

Признак 2,3

- Язык:

Python

- Модель:

Sum of distances of 2, 4 nearest neighbours of each classes

- Признаки:

X

Признак 4

- Язык:

Python

- Модель:

Distances to nearest neighbours of each classes

- Признаки:

Tfidf(X)

Признак 5

- Язык:

Python

- Модель:

Distances to nearest neighbours of each classes

- Признаки:

T-Sne

Признак 6

- Язык:

Python

- Модель:

Clustering features of original dataset

- Признаки:

X

Признак 7

- Язык:

Python

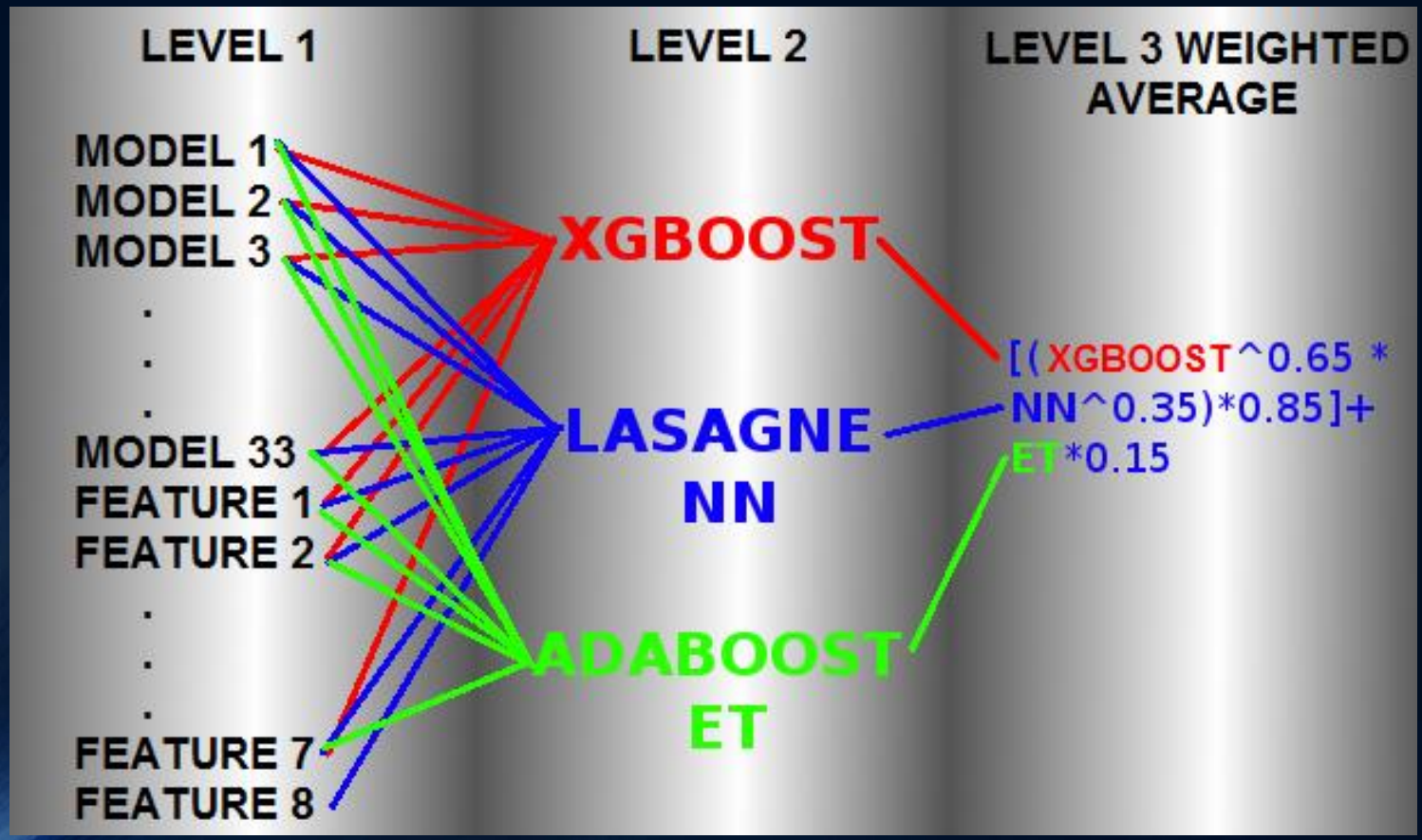
- Модель:

Number of non-zeros elements in each row

- Признаки:

X

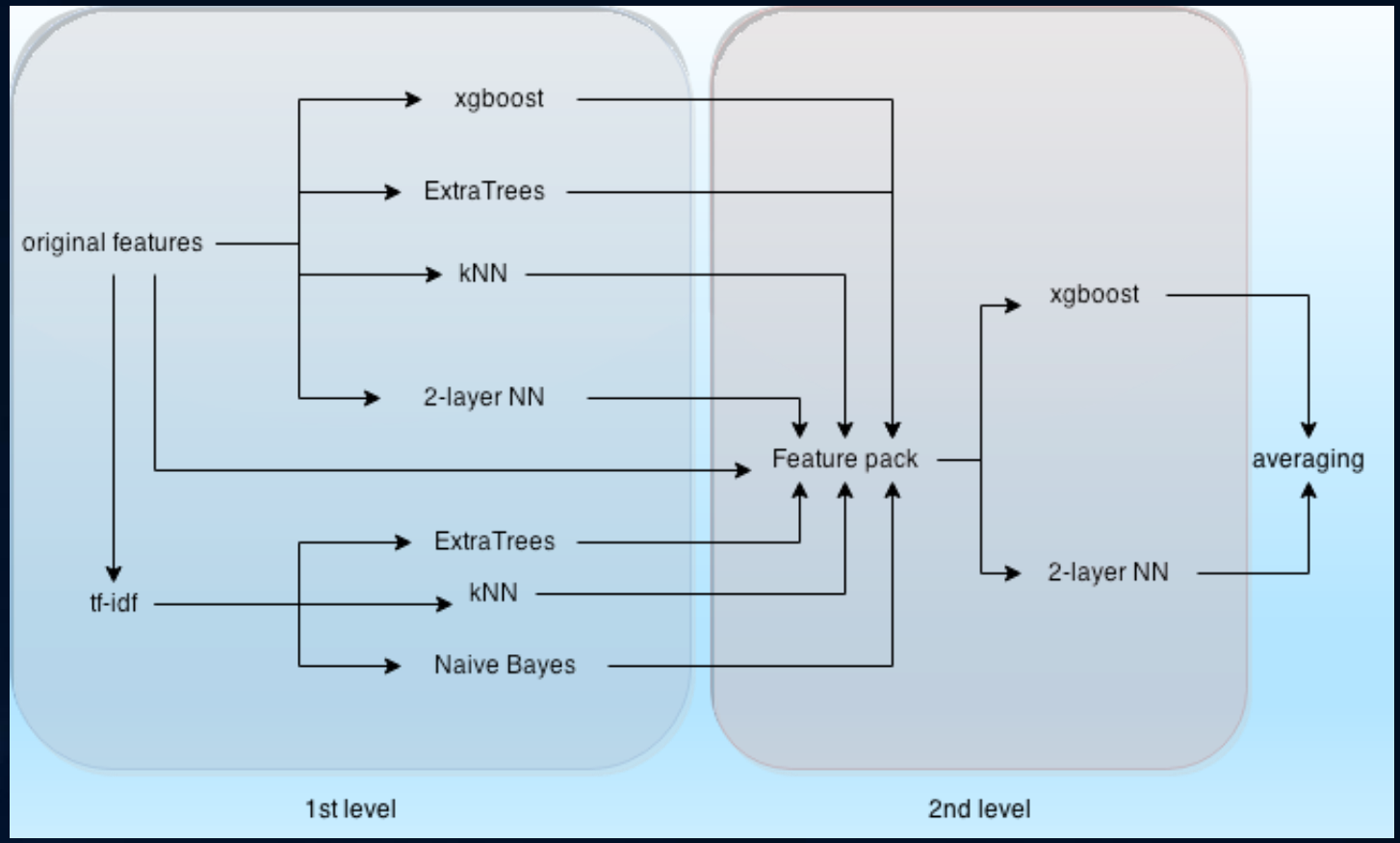
OTTO


LEVEL 1
LEVEL 2
LEVEL 3 WEIGHTED AVERAGE
MODEL 1
MODEL 2
MODEL 3
MODEL 33
FEATURE 1
FEATURE 2
FEATURE 7
FEATURE 8
XGBOOST
LASAGNE NN
ADABOOST ET
[(XGBOOST^0.65 * NN^0.35)*0.85]+ ET*0.15

Итоговое решение

- [XGBOOST^0.65 * NN^0.35] + 0.15 * [ET].

Другие схемы



Следующее занятие

- Use telematic data to identify a driver signature
- https://www.kaggle.com/c/axa-driver-telematics-analysis