CatBoost

4 типа важностей.

Feature importance by Prediction Values Change (Internal Feature Importance) — Internal Feature Importance считается по формуле, в точности равной формуле Prediction Values Change. Разница между важностями в том, что Internal Feature Importance также возвращает важности автоматически добавленных комбинаций на основе категориальных признаков.

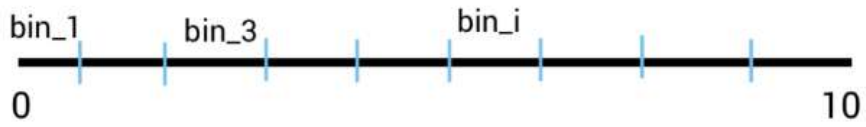
$$feature\_importance_F = \sum_{trees, leafs_F} (v_1 - avr)^2 \cdot c_1 + (v_2 - avr)^2 \cdot c_2,$$

$$avr = \frac{v_1 \cdot c_1 + v_2 \cdot c_2}{c_1 + c_2}, where$$

- $c1, c2$ представляют общий вес объектов в левом и правом листах соответственно. Этот вес равен количеству объектов на каждом листе, если для набора данных не указаны веса.
- $v1, v2$ представляют собой прогнозы модели. $v1$ — это ответ дерева для примеров, для которых выполнено условие разделения, $v2$ — ответ для остальных примеров

Prediction Difference – важность признака для сравнения принятия решений на двух конкретных объектах.

Рассмотрим признак $x10$, разбитый на n частей (бинов). Сделаем прогноз модели для пары объектов, получив $y1, y2$.



1. Для каждого бина от 1 до n
2. Поменяем значение признака x_10 так, чтобы попало в бин (т.е. так что $x10 \in bin_i, \forall i$), получим x'_{10}
3. Вычислим прогноз модели для каждого нового x'_{10}
4. Для каждого нового полученного значения вычислим $difference_{i1} = y1 - cb(x'_{10}), difference_{i2} = y2 - cb(x'_{10})$

После агреггируем полученные значения по всем разбиениям и для пары объектов. Получим среднее изменение прогноза модели, при изменении признака, которое и будет отражать интересующую нас важность.

Loss function change – способ вычисления важности вычисляет разницу между значением потерь модели с этим признаком и без него.

Пусть:

E_iv – математическое ожидание прогноза модели, обученной без i–го признака;
v – вектор со значениями прогноза для исходного набора данных;
$metric$ – это функция потерь, указанная в параметрах обучения.

В зависимости от задачи, вычислим лучшее значение метрики, как:
$bestValue = \pm(metric(E_iv) - metric(v)$

Тогда важность признака по LossFunctionChange есть:

$featureImportance_i = abs(metric(E_iv) - best_value) - abs(metric(v) - best_value)$

XgBoost and LightGbm

3 типа важностей и 2 соответственно.

Cover importnaces — это относительное количество наблюдений, связанных с этим признаком.

Пример:

Пусть у нас есть 100 наблюдений, 6 признаков и 3 дерева. Пусть также признак $feature_1$ используется для определения конечного узла для 13, 5 и 2 наблюдений в $tree_1$, $tree_2$ и $tree_3$ соответственно.

Тогда мы посчитаем $cover_1$ данного объекта, как:

$$cover_1 = 13 + 5 + 2 = 20$$

Это будет рассчитано для всех 6-ти признаков и итоговое покрытие будет равно 20, выраженному в процентах от показателей покрытия всех функций. Таким образом:

$$cover_{fi} = \frac{cover_i}{\sum_i^n cover_n},$$

где n количество признаков. В нашем случае $n = 3$.

Действительно, если просуммировать все покрытия, мы получим значение ≈ 100

Важно: *такая простая интерпретация справедлива только для квадратичной функции потерь (то есть для **линейной регрессии**) Но объяснить данную метрику на классификации проще.*

В случае другой функции потерь это есть сумма градиента второго порядка на обучающих данных, классифицированных благодаря признаку по листьям. Для этого используется Гессиан функции.

Gain importances — важность по приросту. Она аналогична важности, вычисляемой в дереве решений.

В XGB также для gain importances можно получить и полную сумму и нормированную на количество деревьев.
В LightGBM — только полную.

Строится она по такому алгоритму:

- ***Для каждого дерева в ансамбле***
- ***...посчитать вклад каждого признака в чистоту в узлах каждого дерева в модели***
- ***Усреднить по количеству деревьев;***

Более высокое значение этого показателя по сравнению с другим признаком означает, что он более важен для создания прогноза в среднем для деревьев.

Frequence (или weight) importances — представляет собой количество раз, когда конкретный признак встречается в деревьях модели. Разница: в lgbm weight importance называется split importance.

Пусть у нас есть 100 наблюдений, 4 признака и 3 дерева. Пусть также признак $feature_2$ используется для определения конечного узла для *скольких-то* наблюдений в $tree_1$, $tree_2$ и $tree_3$ соответственно и пусть $feature_2$ участвует в 3х, 2х и снова 3х разбиениях в деревьях $1, 2, 3$. Тогда его вес:

$weight = spits_1 + splits_2 + splits_3 = 3 + 2 + 2 = 7.$

	Feature name	Weight importances
0	Pregnancies	102.0
1	Glucose	247.0
2	BloodPressure	180.0
3	SkinThickness	125.0
4	Insulin	102.0
5	BMI	231.0
6	DiabetesPedigreeFunction	268.0
7	Age	188.0

В общем виде вес признака можно записать как:

$$weight = splits_1 + splits_2 + ... + splits_j + ... + splits_n,$$

где n – число дереьев в ансамбле.