

# XGBOOST: Масштабируемая система повышения качества деревьев

ДАУТ БЕКЗАТ.

ГРУППА: АЖ-37.

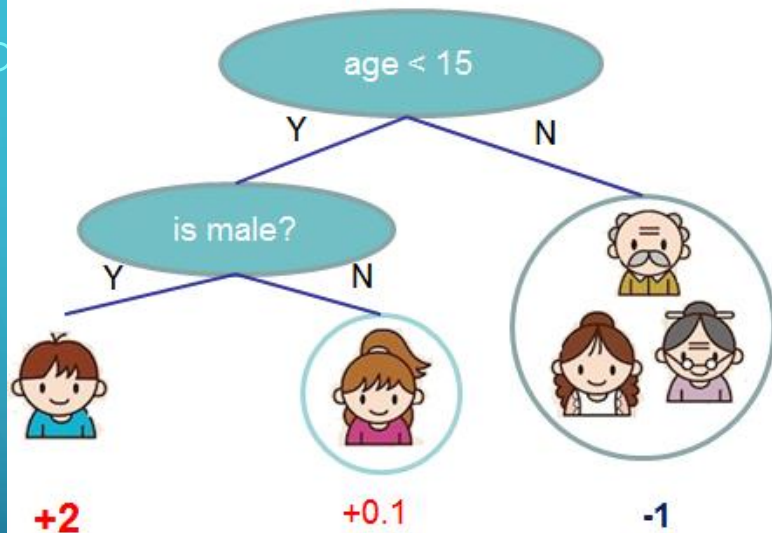
# ЧТО ТАКОЕ XGBOOST?

- **eXtreme Gradient Boosting.**
- Алгоритм контролируемого обучения.
- Вариант модели Gradient Boosted Trees.
  - Поэтапно формирует ансамбль слабых моделей.
  - Сочетает градиентный спуск с CART.

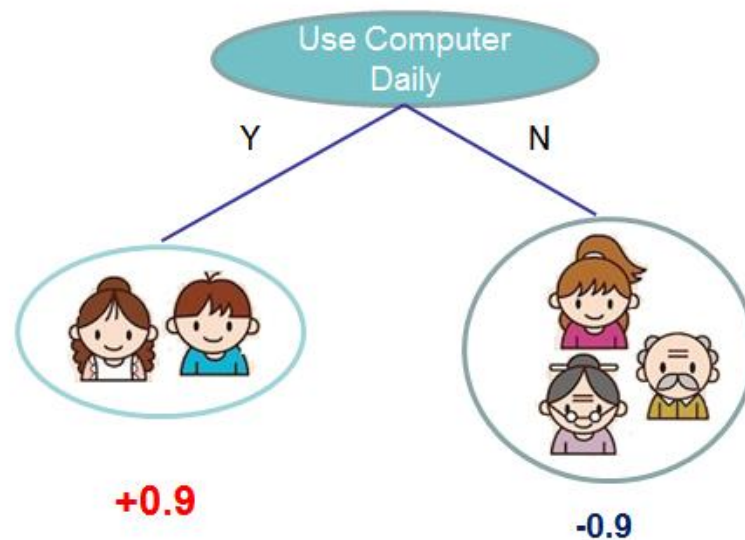
# КОГДА ИСПОЛЬЗОВАТЬ XGBOOST?

- $(y_i, x_i)$   $i$ -й обучающий пример.
- $y_i$  и  $x_i$  может быть непрерывным, категоричным, порядковым ответом и признаками.

tree1



tree2



$f(\text{boy}) = 2 + 0.9 = 2.9$

$f(\text{old man}) = -1 - 0.9 = -1.9$

# ЦЕЛЕВАЯ ФУНКЦИЯ

$$L(\phi) = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(f_k)$$






Функция потерь

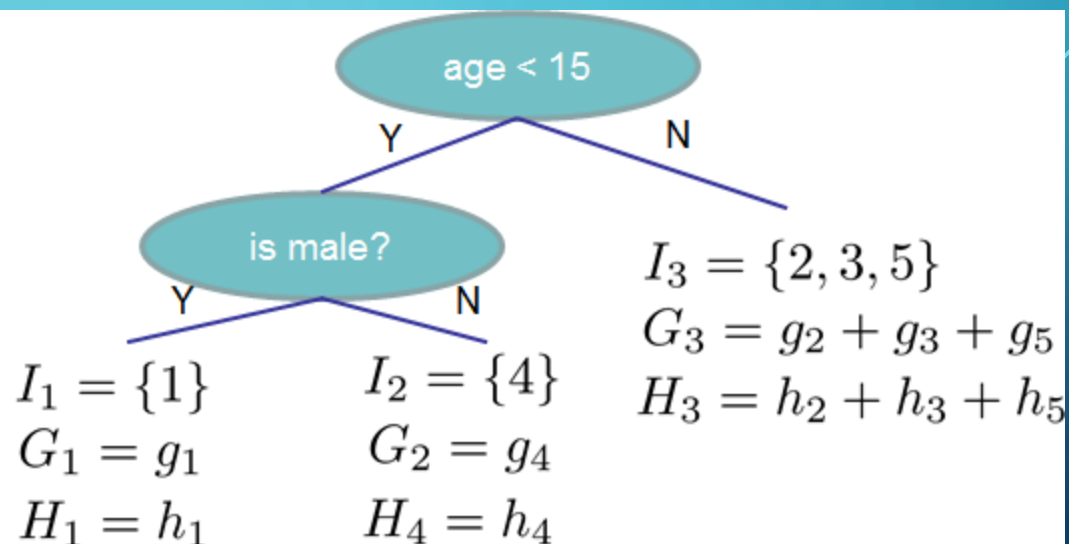
Регуляризация

Дифференцируемая  
выпуклая сложность деревьев

$$\text{где } \Omega(f_k) = \gamma T + \frac{1}{2} \lambda \|w\|^2$$

Instance index      gradient statistics

1		$g_1, h_1$
2		$g_2, h_2$
3		$g_3, h_3$
4		$g_4, h_4$
5		$g_5, h_5$



$$Obj = - \sum_j \frac{G_j^2}{H_j + \lambda} + 3\gamma$$

The smaller the score is, the better the structure is

# ПРИМЕР МОДЕЛИ XGBOOST НА ОСНОВЕ ДАННЫХ ЦВЕТКОВ ВЗЯТЫХ С SCIKIT.

```
import xgboost as xgb
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Load the sample dataset (Iris dataset)
data = load_iris()
X = pd.DataFrame(data.data, columns=data.feature_names)
y = pd.Series(data.target)

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=120)

# Create an XGBoost classifier
clf = xgb.XGBClassifier()

# Train the model
clf.fit(X_train, y_train)

# Make predictions
y_pred = clf.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy * 100:.2f}%")

# Plot feature importance
plt.figure(figsize=(10, 6))
xgb.plot_importance(clf, importance_type='weight', title="Feature Importance (Weight)")
plt.show()
```

# РЕЗУЛЬТАТ РАБОТЫ МОДЕЛИ

