# Важность признаков

Данное задание основано на материалах лекций по логическим методам и направлено на знакомство с решающими деревьями (Decision Trees).

## Вы научитесь:

- обучать решающие деревья
- находить наиболее важные для них признаки

## Введение

Решающие деревья относятся к классу логических методов. Их основная идея состоит в объединении определенного количества простых решающих правил, благодаря чему итоговый алгоритм является интерпретируемым. Как следует из названия, решающее дерево представляет собой бинарное дерево, в котором каждой вершине сопоставлено некоторое правило вида "j-й признак имеет значение меньше b". В листьях этого дерева записаны числа-предсказания. Чтобы получить ответ, нужно стартовать из корня и делать переходы либо в левое, либо в правое поддерево в зависимости от того, выполняется правило из текущей вершины или нет.

Одна из особенностей решающих деревьев заключается в том, что они позволяют получать важности всех используемых признаков. Важность признака можно оценить на основе того, как сильно улучшился критерий качества благодаря использованию этого признака в вершинах дерева.

# Данные

В этом задании мы вновь рассмотрим данные о пассажирах Титаника. Будем решать на них задачу классификации, в которой по различным характеристикам пассажиров требуется предсказать, кто из них выжил после крушения корабля.

# Реализация в Scikit-Learn

В библиотеке scikit-learn решающие деревья реализованы в классах sklearn.tree.DecisionTreeClassifier (для классификации) и sklearn.tree.DecisionTreeRegressor (для регрессии). Обучение модели производится с помощью функции fit.

Пример использования:

```
import numpy as np
from sklearn.tree import DecisionTreeClassifier
X = np.array([[1, 2], [3, 4], [5, 6]])
y = np.array([0, 1, 0])
clf = DecisionTreeClassifier()
clf.fit(X, y)
```

В этом задании вам также потребуется находить важность признаков. Это можно сделать, имея уже обученный классификатор:

```
importances = clf.feature_importances_
```

Переменная importances будет содержать массив "важностей"признаков. Индекс в этом массиве соответствует индексу признака в данных.

Стоит обратить внимание, что данные могут содержать пропуски. Pandas хранит такие значения как nan (not a number). Для того, чтобы проверить, является ли число nan'ом, можно воспользоваться функцией np.isnan.

Пример использования:

```
np.isnan(X)
```

# Материалы

- Подробнее про решающие деревья в sklearn: `http://scikit-learn.org/stable/modules/tree.html`
- Работа с пропущенными значениями в pandas: `http://pandas.pydata.org/pandas-docs/stable/missing_data.html`
- Подробнее о деревьях и их построении: `https://github.com/esokolov/ml-course-msu/blob/master/ML15/lecture-notes/Sem04_trees.pdf`

# Инструкция по выполнению

1. Загрузите выборку из файла titanic.csv с помощью пакета Pandas.
2. Оставьте в выборке четыре признака: класс пассажира (Pclass), цену билета (Fare), возраст пассажира (Age) и его пол (Sex).
3. Обратите внимание, что признак Sex имеет строковые значения.
4. Выделите целевую переменную — она записана в столбце Survived.
5. В данных есть пропущенные значения — например, для некоторых пассажиров неизвестен их возраст. Такие записи при чтении их в pandas принимают значение nan. Найдите все объекты, у которых есть пропущенные признаки, и удалите их из выборки.
6. Обучите решающее дерево с параметром random_state=241 и остальными параметрами по умолчанию.
7. Вычислите важности признаков и найдите два признака с наибольшей важностью. Их названия будут ответами для данной задачи (в качестве ответа укажите названия признаков через запятую без пробелов).

Ответ на каждое задание — текстовый файл, содержащий ответ в первой строчке. Обратите внимание, что отправляемые файлы не должны содержать перевод строки в конце. Данный нюанс является ограничением платформы Coursera. Мы работаем над тем, чтобы убрать это ограничение.