02 - Тема 2 - Лабораторная работа #3 - Деревья решений - Классификатор пола по голосу

Обучить модель машинного обучения классифицировать людей на мужчин и женщин по их голосу? Заметим, чтро системе идентификации пола по голосу, основанной на такой модели, можно найти применение во многих областях — от маркетинга до интегрирования в системы безопасности.

Здесь не нужно работать с сырой аудиоинформацией и преобразовывать её в привычный нам формат табличных данных, поэтому для решения данной используйте готовый датасет, в котором все преобразования аудиоинформации в числовой табличный формат уже произведены. Оригинальную страницу датасета на *Kaggle* с подробным описанием задачи вы можете найти здесь (https://www.kaggle.com/datasets/primaryobjects/voicegender).

Набор данных состоит из 3 168 записанных образцов голоса мужчин и женщин. Образцы предварительно обрабатываются с помощью акустического анализа на языке программирования *R* с использованием специальных библиотек в диапазоне частот 0 Гц-280 Гц (диапазон человеческого голоса). Если вкратце, в результате обработки звук на аудиозаписи оцифровывается и преобразуется в числовую последовательность частот, из которой извлекаются различные статистические характеристики, например средняя частота, с которой говорит спикер на аудиозаписи, или частота с наибольшей энергией и так далее.

Файл с данными `voice_gender.zip`, представленными в формате CSV, вы можете скачать в теме курса.

В столбцах таблицы содержатся статические характеристики для каждой из аудиозаписей, вычисленные на основе акустических свойств.

Важное замечание.

Очень часто в *Data Science* сталкиваются с задачами, которые находятся за гранью области компетенций специалистов из этой области. Типичный пример — медицинские задачи. Датасайентист может не разбираться в том, как получаются те или иные медицинские показатели, в чём они измеряются и какая у них норма, ведь это зона компетенций врача. Однако это же не повод не решать поставленную задачу (хотя, конечно же, базовое понимание внутренней кухни в большинстве случаев даёт определённый бонус).

Мы сейчас как раз рассматриваем пример подобного рода — нам интересен не столько процесс извлечения данных из каждой аудиозаписи, сколько сам результат.

Наша цель состоит в построении модели распознавания пола по уже обработанным аудиозаписям, а не в проектировании процесса их обработки. То есть в процессе решения задачи вы вполне можете абстрагироваться от непосредственного значения самих признаков и воспринимать их в математическом ключе как векторы .

Чуть более подробное описание столбцов.

- `meanfreq` — средняя частота голоса спикера (в кГц);
- `sd` — стандартное отклонение частоты (в кГц);
- `median` — медианная частота (в кГц);
- `Q25` — первый квартиль частоты (25-я квантиль) (в кГц);
- `Q75` — третий квартиль частоты (75-я квантиль) (в кГц);
- `IQR` — межквартильный размах (*Q75-Q25*) (в кГц);
- `skew` — асимметрия распределения частот;
- `kurt` — эксцесс распределения частот;
- `sp.ent` — спектральная энтропия;
- `sfm` — спектральная равномерность;
- `ode` — модальная частота (наиболее популярная частота голоса);
- `centroid` — частотный центроид;
- `peakf` — пиковая частота (частота с наибольшей энергией);
- `meanfun` — среднее значение основной частоты, измеренной по акустическому сигналу;
- `minfun` — минимальное значение основной частоты, измеренной по акустическому сигналу;
- `maxfun` — максимальное значение основной частоты, измеренной по акустическому сигналу;
- `meandom` — среднее значение доминирующей частоты, измеренной по акустическому сигналу;
- `mindom` — минимальное значение доминирующей частоты, измеренной по акустическому сигналу;

- `maxdom` — максимальное значение доминирующей частоты, измеренной по акустическому сигналу;
- `dfrange` — диапазон доминирующей частоты, измеренный по акустическому сигналу;
- `modindx` — индекс модуляции;
- `label` — целевой признак — метка класса: male (голос принадлежит мужчине) или female (голос принадлежит женщине).

Импортируем необходимые библиотеки:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn import tree
from sklearn import model_selection
from sklearn import metrics
```

Итак, приступаем к работе. Прочитаем наши данные:

```
voice_data = pd.read_csv('data/voice.csv')
voice_data.head()
```

	meanfreq	sd	median	Q25	Q75	IQR	skew	kurt	sp.ent	sfm	...	centroid	meanfun	minfun	maxfun	meandom	mi
0	0.059781	0.064241	0.032027	0.015071	0.090193	0.075122	12.863462	274.402905	0.893369	0.491918	...	0.059781	0.084279	0.015702	0.275862	0.007812	0.0
1	0.066009	0.067310	0.040229	0.019414	0.092666	0.073252	22.423285	634.613855	0.892193	0.513724	...	0.066009	0.107937	0.015826	0.250000	0.009014	0.0
2	0.077316	0.083829	0.036718	0.008701	0.131908	0.123207	30.757155	1024.927705	0.846389	0.478905	...	0.077316	0.098706	0.015656	0.271186	0.007990	0.0
3	0.151228	0.072111	0.158011	0.096582	0.207955	0.111374	1.232831	4.177296	0.963322	0.727232	...	0.151228	0.088965	0.017798	0.250000	0.201497	0.0
4	0.135120	0.079146	0.124656	0.078720	0.206045	0.127325	1.101174	4.333713	0.971955	0.783568	...	0.135120	0.106398	0.016931	0.266667	0.712812	0.0

5 rows × 21 columns

Рис. 1: 9308df8a940184c0d8fcdf15bdc21e61.png

Посмотрим на типизацию признаков:

Итак, все признаки, за исключением целевого, кодируются числовым форматом. Типизация целевой переменной не имеет значения для моделей машинного обучения в библиотеке *sklearn*, поэтому кодирование категориальных признаков нам не потребуется.

Заодно проверим данные на наличие пропусков:

```
voice_data.isnull().sum().sum()
# 0
```

Общее количество пропусков в датасете равно 0. Значит, обработка пропущенных значений нам не потребуется.

При желании вы можете провести разведывательный анализ и изучить взаимосвязи между признаками, описывающими голос на аудиозаписи, и целевой переменной, чтобы предварительно определить наиболее значимые признаки и их влияние.

Здесь пропустим этот шаг и перейдём к формированию обучающей и тестовой выборок. Разделим датасет на две части в соотношении 80/20:

```
_# Формируем обучающую и тестовую выборки
X_train, X_test, y_train, y_test = model_selection.train_test_split(X, y,
 ↪  test_size=0.2, stratify=y, random_state=42)
print('Train shape: {}'.format(X_train.shape))
print('Test shape: {}'.format(X_test.shape))
```

Дополнительная предобработка данных нам не потребуется, поэтому мы можем смело перейти к построению моделей. Для моделирования мы, конечно же, будем использовать модели «древесного» типа.

Задание 1. Решающие пни.

```
RangeIndex: 3168 entries, 0 to 3167
Data columns (total 21 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   meanfreq  3168 non-null   float64
 1   sd        3168 non-null   float64
 2   median    3168 non-null   float64
 3   Q25       3168 non-null   float64
 4   Q75       3168 non-null   float64
 5   IQR       3168 non-null   float64
 6   skew      3168 non-null   float64
 7   kurt      3168 non-null   float64
 8   sp.ent    3168 non-null   float64
 9   sfm       3168 non-null   float64
 10  mode      3168 non-null   float64
 11  centroid  3168 non-null   float64
 12  meanfun   3168 non-null   float64
 13  minfun    3168 non-null   float64
 14  maxfun    3168 non-null   float64
 15  meandom   3168 non-null   float64
 16  mindom    3168 non-null   float64
 17  maxdom    3168 non-null   float64
 18  dfrange   3168 non-null   float64
 19  modindx   3168 non-null   float64
 20  label     3168 non-null   object
dtypes: float64(20), object(1)
memory usage: 519.9+ KB
```

Рис. 2: 80b00c864a39edf218401f4c01ee71ba.png

Создайте модель дерева решений максимальной глубины 1 (разумеется, используйте *sklearn*). В качестве критерия информативности используйте энтропию Шеннона.

Обучите модель на тренировочной выборке и визуализируйте её в виде графа.

1. На основе какого фактора будет построено решающее правило в корневой вершине?

- `meanfreq` (средняя частота)
- `median` (медианная частота)
- `IQR` (межквартильный размах частот)
- `meanfun` (средняя основная частота в акустическом спектре)
- `minfun` (минимальная основная частота в акустическом спектре)
- `Q25` (первый квартиль частоты)

2. Чему равно оптимальное пороговое значение для данного фактора? Ответ округлите до трёх знаков после точки-разделителя.
3. Сколько процентов наблюдений, для которых выполняется заданное в корневой вершине условие, содержится в обучающей выборке? Ответ округлите до одного знака после точки-разделителя. Не указывайте в ответе символ %.
4. Сделайте предсказание и рассчитайте значение метрики accuracy на тестовой выборке. Ответ округлите до трёх знаков после точки-разделителя.

Задание 2 Увеличим глубину дерева.

Создайте дерево решений с максимальной глубиной 2. В качестве критерия информативности используйте энтропию Шеннона.

Обучите модель на тренировочной выборке и визуализируйте её в виде графа.

1. Из приведённых ниже факторов выберите те, что используются при построении данного дерева решений:

- A `meanfreq` (средняя частота)
- B `median` (медианная частота)
- C `IQR` (межквартильный размах частот)
- D `meanfun` (средняя основная частота в акустическом спектре)
- E `minfun` (минимальная основная частота в акустическом спектре)
- F `Q25` (первый квартиль частоты)

2. Сколько листьев в построенном дереве содержат в качестве предсказания класс `female`? Для того, чтобы отобразить имена классов при визуализации дерева решения с помощью функции `plot_tree()`, укажите параметр `class_names=dt.classes_`.

- 1
- 2
- 3
- 4

3. Сделайте предсказание и рассчитайте значение метрики accuracy на тестовой выборке. Ответ округлите до трёх знаков после точки-разделителя.

Задание 3 Дадим дереву решений б'ольшую свободу.

Создайте дерево решений, не ограничивая его максимальную глубину. В качестве критерия информативности используйте энтропию Шеннона.

В качестве значения параметра `random_state` возьмите 0.

Обучите модель на тренировочной выборке.

1. Чему равна глубина полученного дерева решения? Глубину дерева можно узнать с помощью метода `get_depth()`.
2. Чему равно количество листьев в полученном дереве решений? Количество листьев можно узнать с помощью метода `get_n_leaves()`.
3. Сделайте предсказание для обучающей и тестовой выборок и рассчитайте значение метрики *accuracy* на каждой из выборок (отдельно на обучающей и тестовой). Ответы округлите до трёх знаков после точки-разделителя.

Задание 4 попробуем найти оптимальные внешние параметры модели дерева решений для поставленной задачи. Воспользуемся классическим методом подбора гиперпараметров — перебором на сетке с кросс-валидацией (`GridSearchCV`).

Примеры кода для изучения: - https://www.kaggle.com/code/fermatsavant/decision-tree-high-acc-using-gridsearchcv - https://www.kaggle.com/code/mabalogun/titanic-gridsearchcv-with-decisiontreeclassifier - https://www.kaggle.com/code/younaniskander/decision-gridsearchcv-accuracy-94-5 - https://www.kaggle.com/code/raisssaid/classification-models-using-gridsearchcv - https://www.kaggle.com/code/satishgunjal/tutorial-k-fold-cross-validation

Задана следующая сетка параметров:

```
_# Задаём сетку параметров
param_grid = {
    'criterion': ['gini', 'entropy'], #критерий информативности
    'max_depth': [4, 5, 6, 7, 8, 9, 10], #максимальная глубина дерева
    'min_samples_split': [3, 4, 5, 10] #минимальное количество объектов,
     ↪  необходимое для сплита
}
```

В качестве кросс-валидатора будем использовать k-fold-валидатор со стратификацией (класс `StratifiedKFold` в *sklearn*):

```
_# Задаём метод кросс-валидации
cv = model_selection.StratifiedKFold(n_splits=5)
```

Другие примеры: - https://www.kaggle.com/code/cpariver/iris-clasification-using-stratified-k-fold - https://www.kaggle.com/code/satishgunjal/tutorial-k-fold-cross-validation

С помощью `GridSearchCV` из модуля `model_selection` библиотеки *sklearn* переберите гиперпараметры дерева решений из приведённой сетки на обучающей выборке и найдите оптимальные. Параметр `random_state` для дерева решений установите равным `0`. В качестве метрики качества (параметр `scoring`) используйте `'accuracy'`.

1. Какой критерий информативности использует наилучшая модель?

- Критерий Джини
- Энтропия Шеннона

2. Чему равна оптимальная найденная автоматически (с помощью `GridSearchCV`) максимальная глубина?

3. Чему равно оптимальное минимальное количество объектов, необходимое для разбиения?

4. С помощью наилучшей модели сделайте предсказание отдельно для обучающей и тестовой выборок. Рассчитайте значение метрики *accuracy* на каждой из выборок. Ответы округлите до трёх знаков после точки-разделителя.

Задание 5 Для оптимального дерева решений, построенного в **задании 4**, найдите важность каждого из факторов (`sklearn.tree.DecisionTreeClassifier.feature_importances_`). Визуализируйте её в виде столбчатой диаграммы.

Выделите топ-3 наиболее важных факторов, участвующих в построении дерева решений:

- A `meanfreq` (средняя частота)
- B `median` (медианная частота)
- C `IQR` (межквартильный размах частот)
- D `meanfun` (средняя основная частота в акустическом спектре)
- E `minfun` (минимальная основная частота в акустическом спектре)
- F `Q25` (первый квартиль частоты)
- F `sfm` (спектральная равномерность)