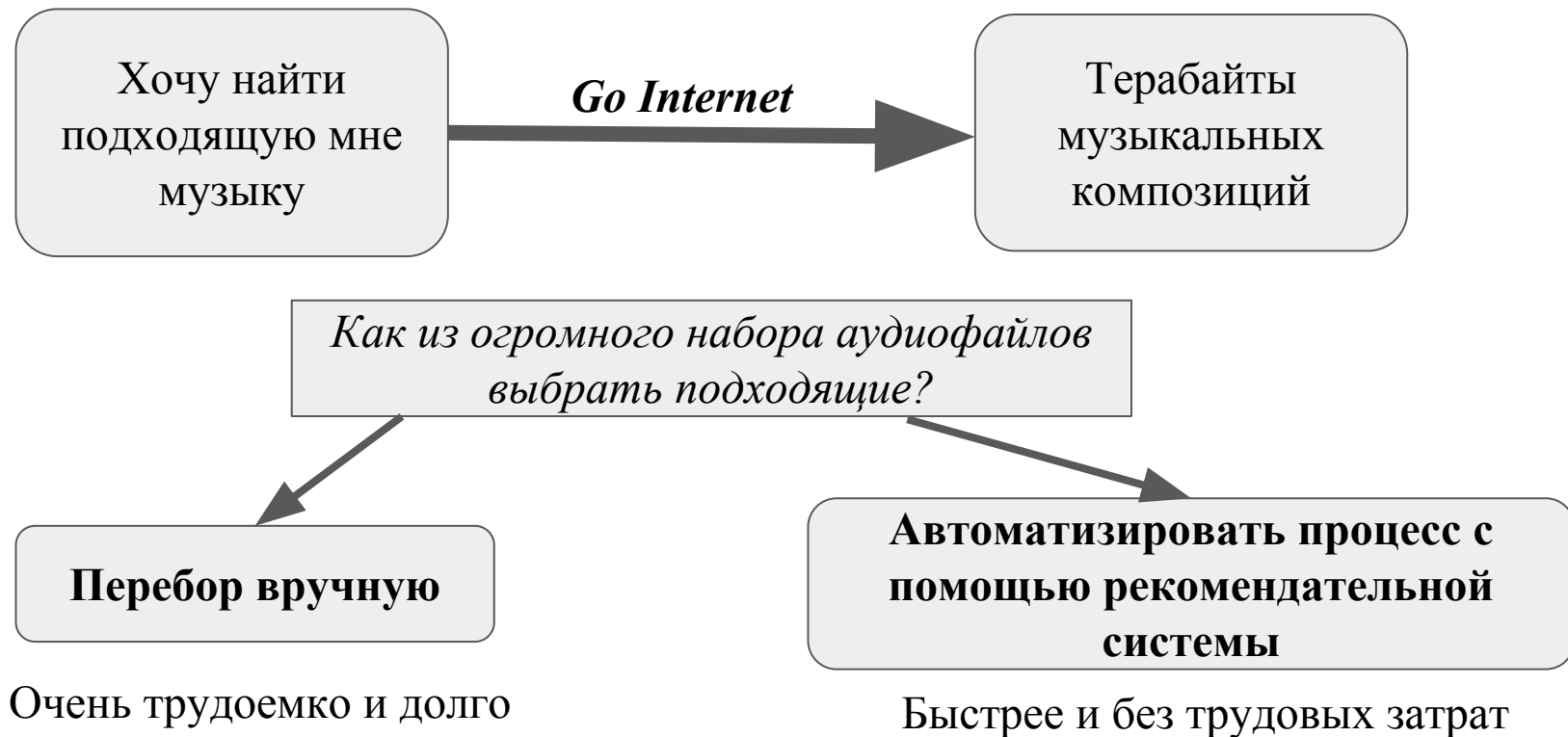


Рекомендательная система музыки

курсовая работа
студента 341 группы
Руденко Дмитрия Андреевича

Введение



Обзор существующих систем



last.fm



Постановка задачи

Целью данной работы является реализация приложения, которое по готовому набору пользовательской музыки формировало бы систему, способную определять, понравится ли конкретный аудиофайл пользователю.



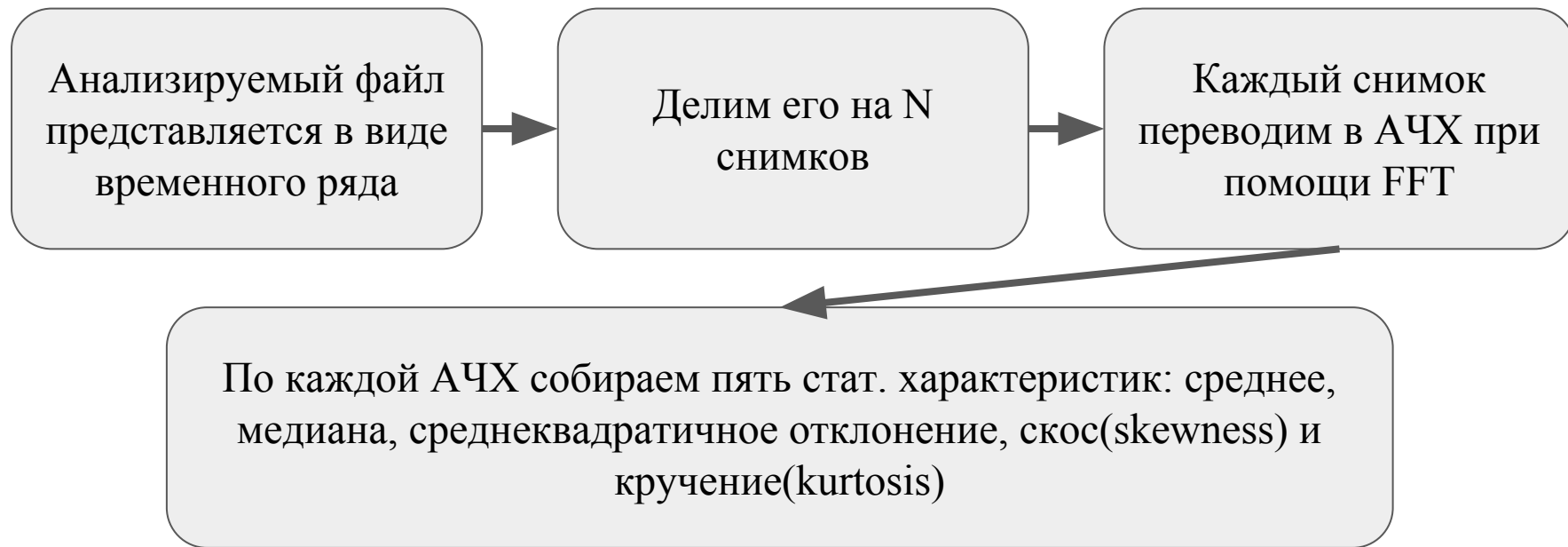
Чтение пользовательского набора

Состоится два набора данных: аудиозаписи, которые нравятся пользователю и которые нет. Чтение реализовано с помощью стандартных библиотек Python*: wave, struct и os.



** Для проекта был выбран язык Python, так как на нем реализованы такие простые в обращении библиотеки, как NumPy и Sklearn, про которые будет рассказано далее.*

Обработка начальных данных



Таким образом, каждая композиция после обработки образует точку в $5N$ - мерном пространстве

Построение рекомендательной модели

Идеи:

1. Построить обобщающий прямоугольник: считаем максимум и минимум по всем координатам, и далее считаем новую композицию подходящей, если каждая из ее координат входит в промежуток между максимумом и минимумом. Очевидный минус: любой значительный выброс ломает модель;

2. Обратиться к алгоритмам классификации Machine Learning: K-Neighbors Classification, Support Vectors Classification, Logistic Regression и Gradient boosting Classification.

Теория

$$X^m = \{(x_1, y_1), \dots, (x_m, y_m)\}.$$

$$\rho(u, x_{1;u}) \leq \rho(u, x_{2;u}) \leq \dots \leq \rho(u, x_{m;u})$$

$$a(u) = \arg \max_{y \in Y} \sum_{i=1}^m [x_{i;u} = y] w(i, u).$$

$$w(i, u) = [i \leq k] \text{ — метод } k \text{ ближайших соседей.}$$

KNeighbors Classification

$$a(x) = \text{sign}(\sum_{j=1}^n w_j f_j(x) - w_0) \quad a(x) = \text{sign}(\sum_{j=1}^n w_j f_j(x) - w_0)$$

$$\sum_{i=1}^{\dots} \ln(1 + \exp(-y_i \langle x_i, w \rangle)) \rightarrow \min_w$$

$$\sum_{i=1}^m (1 - M_i(x, w_0))_+ + \frac{1}{2C} (\|w\|)^2 \rightarrow \min_{w; w_0}$$

$$M_i(x, w_0) = y_i (\langle x_i, w \rangle - w_0)$$

*Logistic
Regression*

Support Vector Classification

Теория

$$X^m = \{(x_1, y_1), \dots, (x_m, y_m)\}.$$

$$a(x) = \sum_{m=1}^T \gamma_m a_m(x)$$

$$a_m = \arg \min_a \sum_{j=1}^N \left(\frac{\partial L(y_j, a_{m-1}(x_j))}{\partial a_{m-1}(x_j)} - a(x_j) \right)^2$$

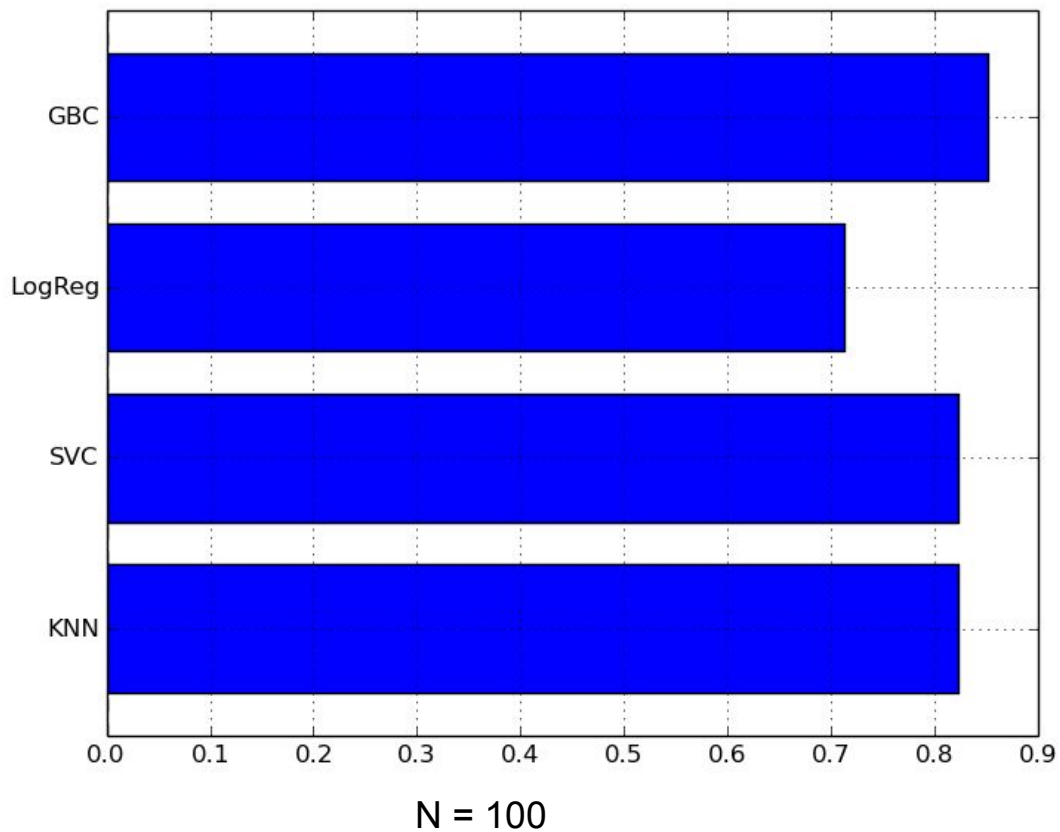
$$\gamma_m = \arg \min_{\gamma} \sum_{j=1}^N L(y_j, a_{m-1}(x_j) + \gamma * a_m(x_j))$$

Gradient boosting Classification

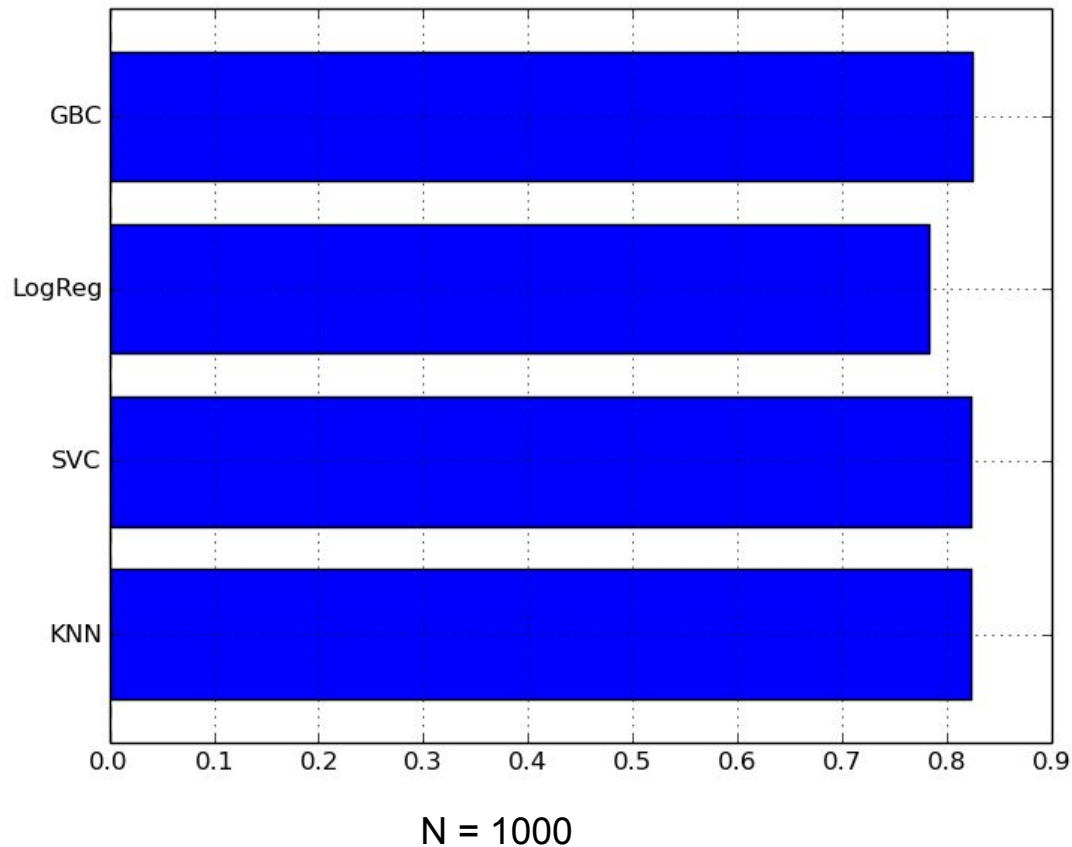
Выборка подходящих композиций из нового набора

Обрабатываем новый набор аналогично обработки начальных данных (собираем статистические данные) и для каждого из объектов запрашиваем у построенной модели ответ на вопрос: "К какому классу принадлежит этот объект?". Те из объектов, которые попадут в класс подходящих ("True"), и будут искомые.

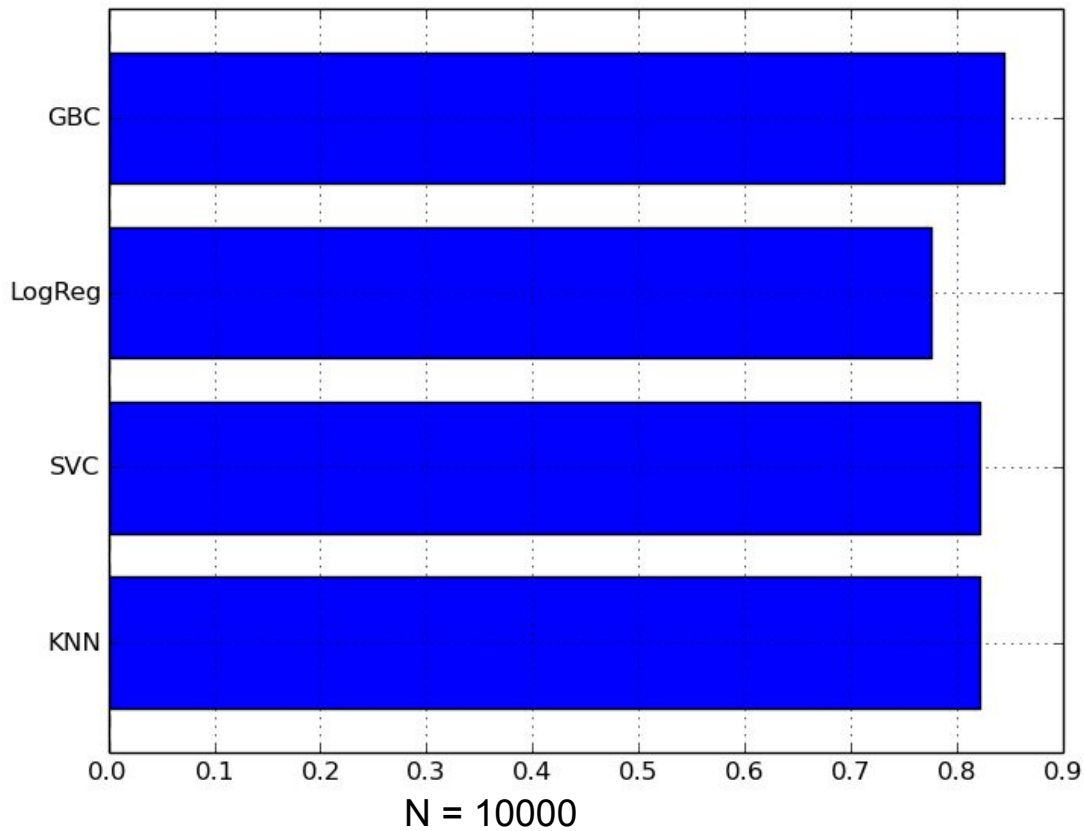
Тестирование



Тестирование



Тестирование



Тестирование

Тестирование на целевом наборе

N	KNC	SVC	LogReg	GBC
100	78.8	75.8	63.6	72.7
1000	72.7	75.6	72.7	72.7
10000	81.8	75.6	75.6	84.8

Заключение

В ходе работы было реализовано приложение, которое по готовому набору пользовательской музыки формирует систему, способную определять, понравится ли конкретный аудиофайл пользователю.

Написаны методы для чтения и обработки аудиофайлов, построения рекомендательной модели, выборки подходящих аудиофайлов из нового набора на языке Python с использованием библиотек NumPy и Sklearn.