# Canary Learning

## Команда

- Никита Шаповалов
  - о Б05-907
  - Преподаватель:
    - Бабичев Сергей Леонидович

- Даниил Лескевич
  - о Б05-907
  - Преподаватель:
    - Бабичев Сергей Леонидович

- Аким Каленюк
  - о Б05-902
  - Преподаватель:
    - Сергей Александрович Гутник

## Проблема



Проблема заключается в том, что владельцам кенаров нужно научить пению какой-либо мелодии самца канарейки, и при этом есть необходимость понимать, насколько хорошо кенар воспроизводит песню.

#### Решение

0

- Учеба кенара на основе методов, описанных в научной статье.
  - Учеба это полное взаимодействие устройства с кенаром
    - В определенное время суток, по согласованию с хозяином, устройство проводит "**семинары**" по пению для кенара
  - **Хозяин** сам **выбирает мелодии**, которые он хотел бы слышать от кенара
- Дополнительная модификация для геймификации обучения:
  - **Соревнования знаниями кенаров** с остальными владельцами;

## Цель курсового проекта.

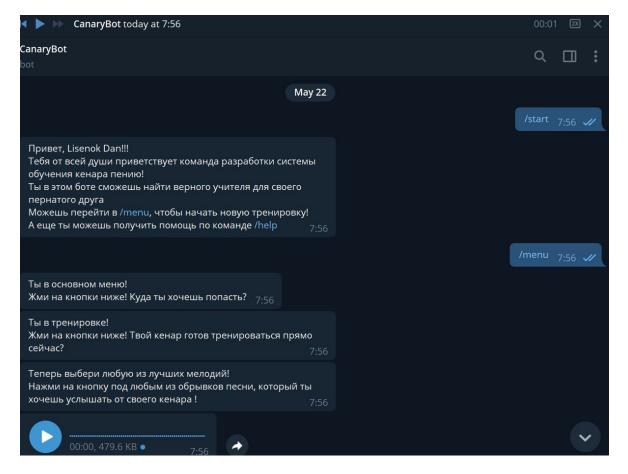
- Целью курсового проекта является разработка программного продукта, позволяющего выполнять обучение кенара пению, анализируя впоследствие его навыки. Для разработки данной программы будет использованы методы, описанные в научных статьях по обучения кенара пению.
- Путь от обычной, «бытовой» постановки задачи до полноценного программного обеспечения, позволяющего решать эту задачу, позволит нам развить компетенции в области перевода любой проблемы, возникающей в нашей жизни, на математический язык, и впоследствии корректно решать поставленную задачу.
- Также целью проекта является получение навыков работы в команде.

#### Основные подзадачи и их взаимосвязь.

В поставленной задаче можно выделить несколько взаимосвязанных подзадач, а именно:

- Построение логики работы программы;
- Создание статистической модели для анализа пения кенара;
- Разработка Телеграм-бота для взаимодействия с хозяином кенара;
- Связывание основных элементов проекта.

## Телеграм-Бот

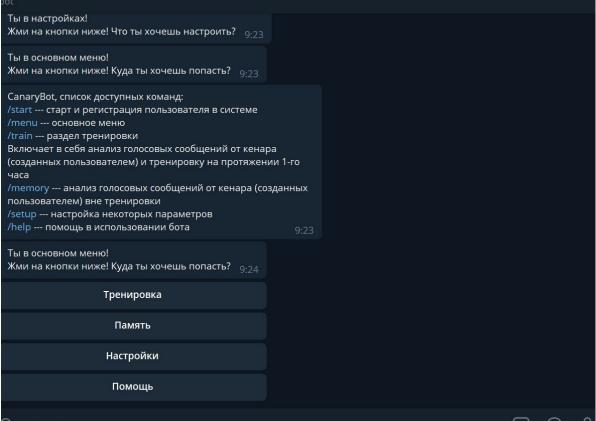


Ты в настройках! Жми на кнопки ниже! Что ты хочешь настроить? 9.73 Ты в основном меню! Жми на кнопки ниже! Куда ты хочешь попасть? 👵 🤈 ২ CanaryBot, список доступных команд: /start --- старт и регистрация пользователя в системе /menu --- основное меню /train --- раздел тренировки Включает в себя анализ голосовых сообщений от кенара (созданных пользователем) и тренировку на протяжении 1-го часа /memory --- анализ голосовых сообщений от кенара (созданных пользователем) вне тренировки /setup --- настройка некоторых параметров /help --- помощь в использовании бота Ты в основном меню! Жми на кнопки ниже! Куда ты хочешь попасть? 9-24 Тренировка Память Настройки Помощь





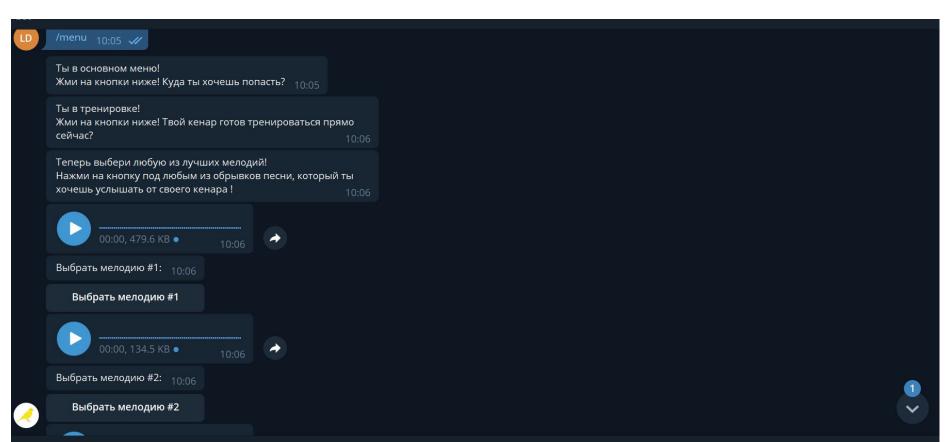


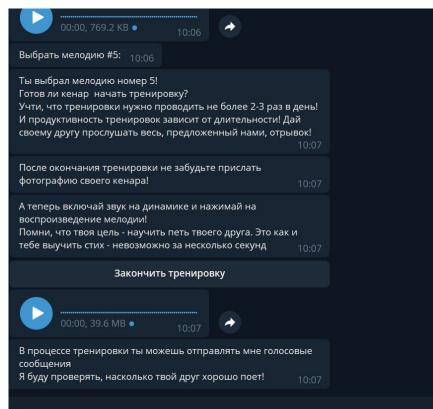










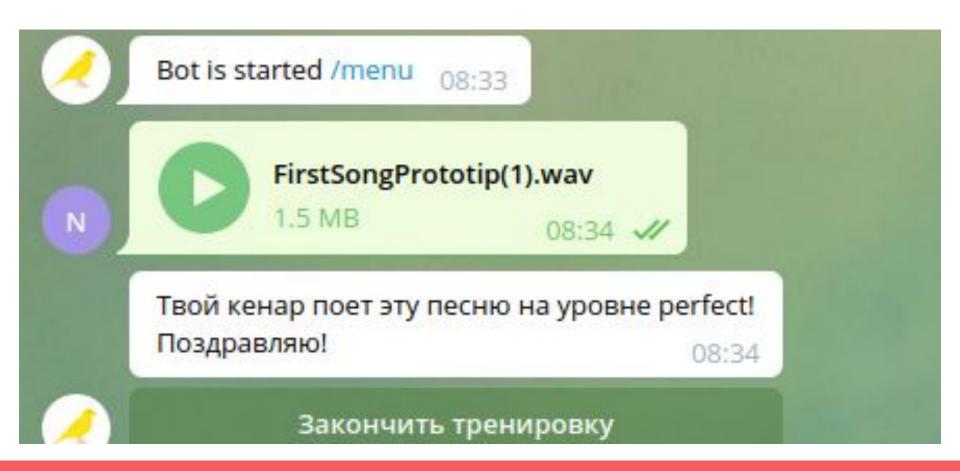








#### Success!



## Структура проекта

```
canary-learning-bot - train.py
File Edit View Navigate Code Refactor Run Tools VCS Window Help
canary-learning-bot audio audio-user
                             ③ Ξ 😤 🗢 — 👸 train.py × 👸 loader.py × 👸 ml_file.py × 🐞 train_buttons.py × 👸 quick_commands.py × 🐞 start.py
   ■ Project ▼
    canary-learning-bot ~/canary-learning-bot
                                                       from PyQt5.QtCore import dec
                                                                                                                                                                                                                A8 A78 × 2 ^ V

✓ ■ audio
                                                       from aiogram import types
      > audio-big
                                                       from aiogram.dispatcher.filters import state
      > audio-small
                                                       from aiogram.dispatcher.filters.builtin import CommandStart, Command, Text
      > 🖿 audio-user
                                                       from aiogram.types import CallbackQuery, InlineKeyboardMarkup, InlineKeyboardButton, InputFile, ContentType
    > 🛅 data
                                                       from aiogram.utils.callback_data import CallbackData
    > DII filters
                                                       import soundfile as sf

→ landlers

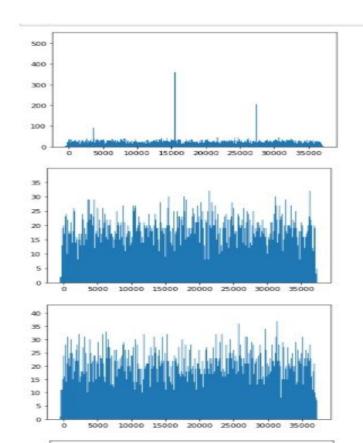
                                                       from filters.private_chat import IsPrivate
      > m channels
                                                       from keyboards.inline.callback_inline_buttons import my_callback
      > 🛅 errors
                                                       from keyboards.inline.menu_buttons import menu_choise
      > 🖿 groups
                                                       from keyboards.inline.setup_buttons import setup_rename_choise
      > users
                                                       from keyboards.inline.train_buttons import train_choise, train_ready1, train_finish, train_finish_choise, train_ready5,
         init .py
                                                           train_ready4, train_ready3, train_ready2
    > kevboards
                                                       from loader import dp, bot
    > middlewares
                                                       from states.train import TrainMelody1
    > la states
                                                       from utils.db api import quick commands as commands
    Y 🛅 utils
                                                       from utils.ml_canary_learning.ml_file import work
      > 🖿 db api
      > 🖿 misc
      > ml canary learning
                                                       @dp.callback_query_handler(Text(["train", "back_to_train"]))
      > 🖿 redis
                                                       async def bot_start(call: CallbackQuery):
         init .py
                                                           await call.answer(cache_time=60)
         is notify admins.py
                                                           await call.message.edit reply markup()
         set bot commands.pv
                                                           await call.message.answer(
       env.
                                                              "\n".join(
       .gitignore
       app.py
                                                                      f'Ты в тренировке!'.
       loader.py
                                                                      f'Жми на кнопки ниже! Твой кенар готов тренироваться прямо сейчас?',
       requirements.txt
                                                                  1),
  > III External Libraries
                                                                  reply_markup=train_choise)
  > Scratches and Consoles
                                                       @dp.message_handler(Command('train'), IsPrivate())
```

# Анализ аудиосообщений

## Сравнение аудиозаписей

- 1. Привести все аудио к wav формату с одинаковым sample rate(использовалась программа bas.sh)
- 2. Применяем преобразование Фурье, после чего оставляем только самые сильные частоты.(отпечаток)
- 3. Сравниваем отпечатки(находим момент с совпадением).
- 4. Применяем перестановочный критерий начиная с момента с максимальным совпадением.

## Сравниваем отпечатки 1

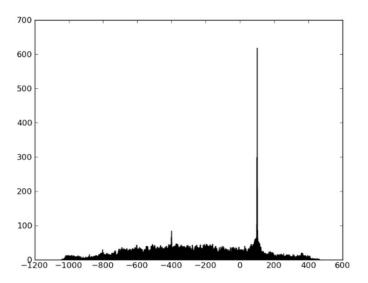


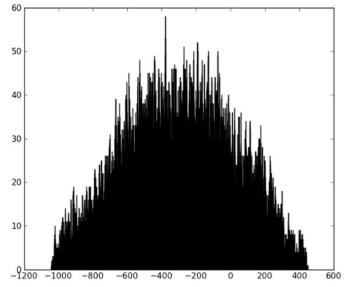
Можете попробовать догадаться на каком из графиков в мы нашли отпечаток нашей песни.

#### Сравниваем отпечатки 2

Понятно что отпечатки никогда не совпадут в точности из-за шумов и искажений. Но оказывается что таким огрубленные (в пункте 2) частоты достаточно хорошо переживают все искажения, и достаточно большой процент частот совпадает в точности — таким образом, нам остается только найти сдвиг при котором среди двух последовательностей частот много

совпадений.





## Перестановочный критерий 1

выборки:

нулевая гипотеза:

альтернатива:

статистика:

нулевое распределение:

$$X_1^{n_1} = (X_{11}, \dots, X_{1n_1}),$$

$$X_2^{n_2} = (X_{21}, \dots, X_{2n_2}),$$

$$H_0: F_{X_1}(x) = F_{X_2}(x);$$

$$H_1: F_{X_1}(x) = F_{X_2}(x + \Delta), \Delta < \neq > 0;$$

$$T(X_1^{n_1}, X_2^{n_2}) = \frac{1}{n_1} \sum_{i=1}^{n_1} X_{1i} - \frac{1}{n_2} \sum_{i=1}^{n_2} X_{2i};$$

порождается перебором  $C_{n_1+n_2}^{n_1}$ 

размещений объединённой выборки.

## Перестановочный критерий 2

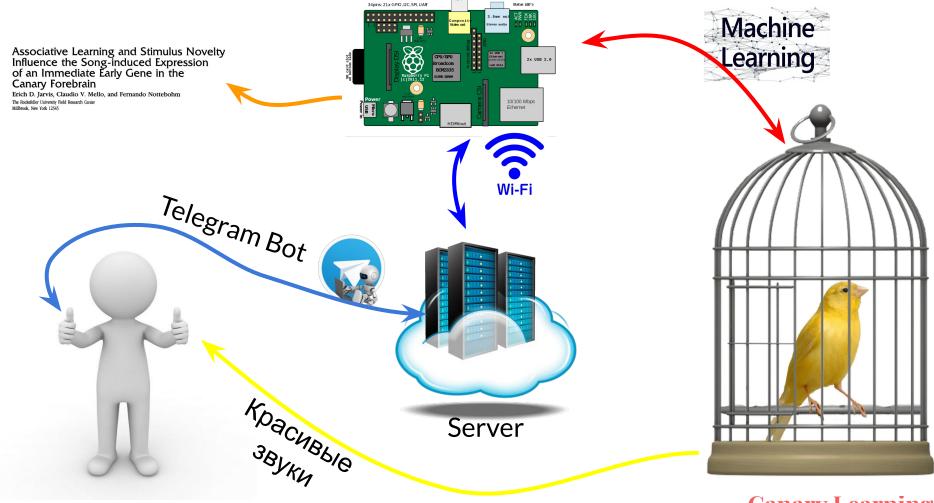
Мы применяем перестановочный критерий в соответствии с теорией с предыдущего слайда.

За X1, X2 принимаются данные аудиозаписей начиная с максимального совпадения.

#### Использованные Библиотеки

```
import os
   import numpy as np
   import numpy as np
5
   import pandas as pd
6
   import itertools
   from scipy import stats
   from statsmodels.stats.descriptivestats import sign test
   from statsmodels.stats.weightstats import zconfint
10
11
   from statsmodels.stats.weightstats import *
12
13
   import numpy
   from matplotlib import pyplot, mlab
14
   import scipy.io.wavfile
15
   from collections import defaultdict
16
```

# Будущее тут!



3 polesjack

**Canary Learning** 

Machine Associative Learning and Stimulus Novelty Influence the Song-induced Expression of an Immediate Early Gene in the Learning Canary Forebrain Erich D. Jarvis, Claudio V. Mello, and Fernando Nottebohm CIOFCUD II способ реализации Kpachbble 3Bykh Server

**Canary Learning**