

Qaz Letters

Kazakh - Qaz Letters

Қазақ әріптері

Қазақ тілі 42 таңбадан тұратын кирилл әліпбиін қолданады, оның ішінде тоғыз ерекше әріп бар. Қазіргі компьютерлік көру әдістері көбінесе көлемді, соңы біртұтас терең нейрондық желілерге сүйенсе, бұл жарыста басты назар — ерекшелік құрастыру (feature engineering) мен классикалық машиналық оқыту (ML) әдістерін қолдануға аударылады.

Қатысушылар өңделмеген кескіндермен жұмыс істемейді. Олардың орнына әр қазақ әрпінің бейнесі бойынша визуалды ақпарат мұқият дайындалған, сандық сипаттамаларға түрлендірілген кестелік деректер жиыны беріледі.

Сипаттама

Сіздің міндетіңіз — кестелік деректер жиынын пайдалана отырып, классификация моделін құру. Бұл деректер жиынында келесі толық сипаттамалар бар:

Статистикалық белгілер:

- Орташа қарқындылық: пиксельдердің орташа мәні.
- Стандартты ауытқу: пиксель мәндерінің таралуы.
- Минималды/максималды қарқындылық: экстремалды мәндерді анықтауға көмектеседі.
- Асимметрия/экссесс (skewness/kurtosis): қарқындылықтың асимметриясын және өткірлігін өлшейді.
- Пиксель мәндерінің гистограммасы: қарқындылықты интервалдарға (мысалы, 16 бөлікке) бөліп, олардың жиілігін есептейді.

Пішінге / геометрияға негізделген белгілер:

- Центроид: таңбаның масса орталығы.
- Шектеуші тікбұрыш (bounding box): ені, биіктігі және жақтар қатынасы.
- Пиксель тығыздығы: нөлге тең емес пиксельдердің үлесі.
- Шекара пиксельдерінің саны: шекараларды анықтау әдісімен алынған пиксельдер саны.
- Моменттер / Ху моменттері: пішіннің айналу мен масштабқа тәуелсіз 7 инвариантты сипаттамасы.
- Контурлар: контурлардың саны немесе олардың аудандары (OpenCV арқылы алынған).
- Радиалды белгілер: нөлге тең емес пиксельдердің центроидқа дейінгі қашықтықтары.
- Фурье-дескрипторлар: айналу мен масштабқа тәуелсіз пішін сипаттамалары.
- Симметрия өлшемдері: сол/оң немесе үстіңгі/астыңғы бөліктердің ұқсастығын салыстыру.

Проекциялық / профильдік белгілер:

- Көлденең және тік проекциялар: пиксель мәндерінің жолдар мен бағандар бойынша қосындысы.
- Аймақтық белгілер (zoning): кескінді аймақтарға бөлу (мысалы, 4×4) және әр аймақтағы пиксель тығыздығын есептеу.

Түрлендіру негізіндегі белгілер:

- Фурье түрлендіруі: жиілік аймағындағы коэффициенттер.
- Дискретті косинустық түрлендіру (DCT): көбіне сығу әдістерінде қолданылады, төмен жиілікті үлгілерді анықтайды.
- Вейвлет түрлендіруі: көпмасштабты ыдырату, шеттер мен текстураларды сипаттайды.

Бағалау

Қатысушылардың шешімдері Assurance метрикасы бойынша бағаланады — жасырын тест жиынындағы дұрыс анықталған таңбалардың үлесі.

Файлдар

- **train.csv** — оқыту деректер жиыны. Әріп кескіндерінен алынған жасанды белгілерді (мысалы, f1, f2, ...) және нақты таңба меткасын (target) қамтиды.
- **test.csv** — тест деректер жиыны. Сол жасалған белгілер жиынтығын қамтиды, бірақ мақсатты баған (target) болмағандықтан, оны болжау қажет.
- **sample_submission.csv** — дұрыс пішімдегі үлгілік жіберу файлы.
- **labels.csv** — деректер туралы қосымша ақпарат.

Russian - Qaz Letters

Казахские буквы

Казахский язык использует кириллический алфавит, состоящий из 42 символов, включая девять уникальных букв. Хотя современные методы компьютерного зрения часто опираются на модели глубокого обучения, в данном соревновании внимание уделяется искусству построения признаков и применению классических методов машинного обучения (ML).

Участники не будут работать с необработанными изображениями, а получат специально подготовленный табличный набор данных, где визуальная информация о каждой букве казахского алфавита преобразована в набор тщательно разработанных числовых признаков.

Описание

Ваша задача — построить классификационную модель, используя табличный набор данных. Этот набор данных содержит полный набор признаков:

Статистические признаки:

- Средняя интенсивность: среднее значение пикселя.
- Стандартное отклонение: разброс значений пикселей.
- Минимальная/максимальная интенсивность: помогает захватывать экстремумы.
- Асимметрия/эксцесс (skewness/kurtosis): измеряют асимметрию и остроту распределения интенсивностей.
- Гистограмма значений пикселей: разделение интенсивности на интервалы (например, 16 бинов) и подсчёт вхождений.

Признаки, основанные на форме / геометрии:

- Центроид: центр масс символа.
- Ограничивающий прямоугольник: ширина, высота и соотношение сторон.
- Плотность пикселей: доля ненулевых пикселей.
- Количество граничных пикселей: число пикселей границы, получаемое методом детекции границ.
- Моменты / моменты Ху: 7 инвариантных моментов, описывающих форму, инвариантных к повороту и масштабу.
- Контурные: число контуров или площади контуров, извлечённые с помощью OpenCV.
- Радиальные признаки: расстояния ненулевых пикселей от центроида.
- Фурье-дескрипторы контуров: дескрипторы формы, инвариантные к вращению/масштабу.
- Меры симметрии: сравнение левой-правой или верхней-нижней половин.

Проекционные / профильные признаки:

- Горизонтальные и вертикальные проекции: сумма значений пикселей по строкам и столбцам.
- Зонирование (zoning): деление изображения на зоны (например, 4×4) и вычисление плотности в каждой зоне.

Преобразовательные признаки:

- Преобразование Фурье: коэффициенты в частотной области.
- Дискретное косинусное преобразование (DCT): широко используется в сжатии, захватывает низкочастотные паттерны.
- Вейвлет-преобразование: многомасштабное разложение, захватывающее края и текстуры.

Оценка

Решения участников будут оцениваться по метрике **Accuracy** — доле правильно предсказанных символов на скрытом тестовом наборе данных.

Файлы

- **train.csv** — обучающая выборка. Содержит построенные признаки (например, f1, f2, ...), полученные из изображений букв, а также истинную метку символа (target).
- **test.csv** — тестовая выборка. Содержит тот же набор построенных признаков, но без столбца с целевой переменной (target), который нужно предсказать.
- **sample_submission.csv** — пример файла отправки в правильном формате.
- **labels.csv** — дополнительная информация о данных.

Who best pitcher

Kazakh - Who is the Best Pitcher?

Кім ең мықты питчер?

Әділхан  мен Дархан  - спортты, әсіресе бейсболды қатты жақсы көретін екі дос.

Олар жанкүйерлер өз ойларын жазып, сұрақ қойып, ойындар туралы пікір алмаса алатын **BaseballAndMath.com** деген сайт құрды.

Көп ұзамай бұл сайт әлемнің түкпір-түкпірінен бейсбол статистикасы мен ойыншылары жайлы қызу талқылаулар өтетін қауымдастыққа айналды.

Басында бәрі жақсы жүрді — адамдар өздерінің сүйікті командалары мен ойыншылары жайлы жазды, қауымдастық тез өсе бастады.


Бірақ Әділхан мен Дархан үлкен мәселені байқады:

пайдаланушыларға нақты уақытта спорт статистикасына қатысты сұрақтарға жауап керек, мысалы:

- «Отанидің осы маусымда неше хоум-раны бар?»
- «Кеше “Янкис” үшін кім ұпай жинады?»

Олар **ChatGPT**, **Gemini** және басқа танымал жасанды интеллект құралдарын қолданып көрді. Бұл модельдер әсерлі болғанымен, бір маңызды шындық анықталды:

 ЖИ құралдар **нақты уақыттағы спорт деректерін өңдей алмады.**


 Олар жақында болған оқиғалар мен статистика туралы ескі немесе қате ақпарат береді.

Сол сәтте екі дос шешім қабылдады:

«Егер мұндай құрал жоқ болса, өзіміз жасайық!»

Сөйтіп Әділхан мен Дархан нақты деректерге сүйенетін, **спорт туралы сұрақтарға жауап беретін ЖИ “agent”** құруға кірісті.

Олардың мақсаты айқын болды:

 **MLB (Major League Baseball)** бойынша сұрақтарға дәл әрі жаңартылған жауап беретін **ЖИ agent** жасау.

Неліктен бұл тәсіл тиімді: RAG

Олардың шешімі **Retrieval-Augmented Generation (RAG)** әдіспен жасауға болады.

RAG деген не?

RAG екі күшті тәсілді біріктіреді:

1. **Іздеу (Retrieval)** – мәліметтер базасынан, API-ден ең жаңа ақпаратты табу.
2. **Генерация (Generation)** – LLM(large language model) арқылы сұрақты түсініп, жауап құрастыру.

Модель тек өз ішіндегі ақпаратқа (ол ескірген болуы мүмкін) сүйенудің орнына, RAG әдіспен жауап береді.

✓ **Дәл** – нақты статистика мен жаңа деректер

✓ **Жаңартылған** – соңғы спорт оқиғалар

✓ **Пайдалы** – адамға түсінікті түрде берілген

✨ Бұл жарыстағы мақсат:

Әділхан мен Дарханға MLB жанкүйерлеріне арналған ЖИ agent жасауға көмектес!

Сипаттама

Бұл байқауда **MLB** (Major League Baseball) туралы сұрақтарға жауап бере алатын агент құру керек. Агент бірнеше күрделі **JSON** форматындағы деректермен логикалық жұмыс істей алуы керек.

1. Деректерді жүктеу және талдау:

Үлкен JSON форматындағы деректеді тиімді түрде жүктеп, қалыпқа келтіру.

2. Білімге қол жеткізу:

Әр сұраққа қажетті ең маңызды деректерді тез табу үшін іздеу және индекстеу жүйесін жобалау.

3. Түсіну және жауап беру:

MLB туралы сұрақтарға (ойыншылар және т.б.) нақты әрі дұрыс жауаптар құрастыру.

4. Жалпылау.

Бағалау

Бағалау нәтижесі **дәлдікке (accuracy)** негізделеді — яғни дұрыс жауаптар санының жалпы сұрақтар санына қатынасы арқылы есептеледі.

Жіберілетін файл (Submission File)

Әрбір тест ID үшін жауапты көрсету қажет.

ID,ANSWER

1,"Paris"

2,"42"

3,"blue"

- ID — сұрақтың бірегей идентификаторы.
- ANSWER — жол (string) түрінде беріледі.

Маңызды:

- ANSWER мәні **әрқашан жол (string)** форматында болуы керек.
 - Егер бастапқы JSON дерегінде сан (бүтін немесе қалқымалы) болса да, оны жол түрінде сақтау қажет.
 - Мысалдар:
 - $1 \rightarrow "1"$
 - $1.00 \rightarrow "1.00"$



Дәл сәйкестік (Exact Match)

Жауаптың мағынасын еш өзгертуге болмайды.

ANSWER дәл сол күйінде, JSON файлындағы мәнмен **бірдей болуы керек**.

- қысқартуға немесе форматты өзгертуге **болмайды**.

Мысал:

- JSON мәні: 123.4500
 -  Дұрыс жауап: "123.4500"
 -  Қате жауаптар: "123.45" немесе "123.45 (шамамен)"

Датасет

Бұл деректер жиыны **Major League Baseball (MLB)** арналған бірнеше **endpoint**-тен жиналған біріктірілген JSON файлдар.

-
- **players.json** – ойыншылар туралы біріктірілген деректер
 - **teams.json** – командалар туралы біріктірілген деректер
 - **league.json** – лига деңгейіндегі біріктірілген деректер
-

Барлық деректер **JSON форматында** сақталған.

Жоғарғы деңгейдегі кілттер – бұл объектінің атауы (мысалы, ойыншының немесе команданың аты).

Мысал құрылымдар

- **players.json**
 - Негізгі кілт: ойыншының аты
 - Мәні: барлық ойыншылар үшін ортақ атрибуттардан тұратын сөздік (dictionary)
- **teams.json**

- Негізгі кілт: команданың аты
 - Мәні: барлық командалар үшін бірдей атрибуттар жиыны
 - **league.json**
 - Жоғарғы деңгейде (root жоқ): метадеректер, командалар, турнир кестесі және үздік ойыншылар тізімі (leaders)
-

Бағалау файлдары

test.csv

- **Бағандар:** ID, question
 - **Сипаттама:** Әрбір question(сұрақ) — players.json, teams.json және league.json деректерге негізделіп құрылған.
 - **Мақсат:** Әр ID үшін осы JSON ақпаратқа сүйеніп **ANSWER** (жауап) табу.
-



submission.csv

- **Бағандар:** ID, ANSWER
- **Түсіндірме:**
 - test.csv ішіндегі әрбір ID үшін бір жолдан тұруы керек.
 - Файлда **5 кездейсоқ жолда** дұрыс ANSWER бар.
 - Мұндай бастапқы файлды жіберу арқылы сенің инференс жүйең дұрыс жұмыс істеп тұрғанын тексеруге және **нөлден жоғары ұпай алуға** болады.

Russian - Who is the Best Pitcher?

Кто самый лучший питчер?

Обзор

Адилъхан  и Дархан  — два друга, которые любят спорт, особенно бейсбол.

Они запустили публичный сайт — **BaseballAndMath.com**, где болельщики могли писать посты, задавать вопросы и делиться мыслями об играх.

Он быстро превратился в живое сообщество для обсуждения счетов, игроков и бейсбольных фактов со всего мира.

Сначала всё шло хорошо — люди писали о любимых командах и игроках, и сообщество росло.

Но вскоре Адильхан и Дархан заметили большую проблему:

пользователям нужны были **ответы в реальном времени** на вопросы о спортивной статистике, например:

- «Сколько хоум-ранов Отани выбил в этом сезоне?»
- «Кто набирал очки за “Янки” вчера?»

Они попробовали популярные ИИ-инструменты — **ChatGPT**, **Gemini** и другие. Модели впечатляли, но вскоре стало ясно:

 ИИ-агенты **не справлялись с данными в реальном времени по спорту.**


 Они выдавали устаревшие или неверные ответы о недавних событиях и статистике.

Тогда друзья решили:

«Если ни один инструмент это не умеет, сделаем свой!»

Так Адильхан и Дархан взялись создавать **ИИ-агента для спортивного Q&A**, который опирается не только на статические знания, но и на **живые, реальные данные** из проверенных источников.

Их миссия стала понятной:

 Построить **агента на базе LLM**, который сможет точно отвечать на вопросы о Major League Baseball (MLB), даже когда игры идут и статистика обновляется каждую секунду.

Почему это работает: RAG

Решение основано на **Retrieval-Augmented Generation (RAG)**.

Что такое RAG?

RAG объединяет сильные стороны двух миров:

1. **Retrieval (извлечение)** — получение наиболее релевантной, актуальной информации из базы данных, API или другого источника знаний.
2. **Generation (генерация)** — использование большой языковой модели (LLM) для понимания вопроса и формирования естественного, полезного ответа на базе извлечённых фактов.

Вместо того чтобы полагаться только на «память» LLM (которая может быть устаревшей), RAG обеспечивает, что ответы:

- ✅ **Точные** — опираются на реальные статистические данные
- ✅ **Актуальные** — отражают последние спортивные события
- ✅ **Полезные** — подаются в виде естественных, понятных объяснений

Ваше задание

✨ Задача соревнования:

Помогите Адильхану и Дархану построить их RAG-агента для фанатов MLB!

Описание

Это соревнование ставит перед вами задачу создать агента вопрос–ответ для MLB, который умеет рассуждать по большим структурированным JSON-датасетам, агрегированным из нескольких эндпоинтов.

Что вы построите

- **Загрузка и парсинг:** эффективно загружать и нормализовать большие JSON-файлы (несколько эндпоинтов, гетерогенные схемы).

- **Слой доступа к знаниям:** спроектировать механизмы lookups/индексы для извлечения минимально релевантных фактов под каждый запрос.
- **Рассуждение и ответы:** выдавать точные, хорошо отформатированные ответы на вопросы пользователей об MLB (команды, игроки, матчи, сезоны, рекорды и т. д.).
- **Обобщение:** надёжно работать на скрытом валидационном наборе вопросов.

Оценка

Оценка решений производится по **accuracy (точности)** — числу правильных ответов, делённому на общее число вопросов.

Файл отправки (Submission)



Для каждого ID в тестовом наборе необходимо указать предсказанный ответ. Файл должен содержать заголовок и иметь следующий формат:

```
ID,ANSWER
1,"Paris"
2,"42"
3,"blue"
etc.
```

- ID — идентификатор вопроса.
- ANSWER — строка.

Требования к полю ANSWER:

- Поле **всегда** должно быть строкой.
 - Даже если исходное значение в JSON — число (целое или с плавающей точкой), его нужно сохранить как строку.
 - Примеры:
 - 1 → "1"
 - 1.00 → "1.00"
- **Точное совпадение (Exact Match):**

- Будьте внимательны: нельзя преобразовывать или переформатировать значение.
- ANSWER должен **в точности** совпадать с исходным значением из JSON.
- **Нельзя** округлять, обрезать или менять форматирование.
- Пример:
 - Значение в JSON: 123.4500
 -  Корректно: "123.4500"
 -  Некорректно: "123.45" или "123.45 (approx.)"

Датасет

Этот датасет содержит агрегированные данные из эндпоинтов в формате JSON для Major League Baseball (MLB). Каждый JSON-файл соответствует определённому типу сущности с согласованной структурой ключей для всех объектов.

Файлы

- **players.json** – агрегированные данные на уровне игроков
- **teams.json** – агрегированные данные на уровне команд
- **league.json** – агрегированные данные на уровне лиги

Формат и структура

Каждый файл хранится в формате **JSON**. Ключи верхнего уровня — это имена сущностей (например, имя игрока или команды). Под каждым корневым ключом у сущности задан единый набор атрибутов.

Примеры структур

- **players.json**
 - Корневой ключ: имя игрока

- Значение: словарь с одним и тем же набором атрибутов для каждого игрока
- **teams.json**
 - Корневой ключ: название команды
 - Значение: словарь с одним и тем же набором атрибутов для каждой команды
- **league.json**
 - Верхний уровень (без корневого ключа league): метаданные, команды, турнирные таблицы (standings), лидеры (leaders)

Файлы для оценки

test.csv

- **Столбцы:** ID, question
- **Описание:** Каждый question сгенерирован на основе данных из players.json, teams.json и league.json.
- **Цель:** Для каждого ID сгенерировать **ANSWER**, используя ваш алгоритм вывода по этим JSON-источникам.

submission.csv

- **Столбцы:** ID, ANSWER
- **Требования:**
 - Должна быть ровно одна строка на каждый ID из test.csv.
 - Ровно **5 случайных строк** содержат корректный ANSWER.
 - Все остальные строки содержат строку **“no answer”**.
 - Отправка этого базового файла гарантирует **ненулевой результат**, что подтверждает работоспособность вашего кода.

Missing Fundamental Puzzle

Kazakh - Missing Fundamental Puzzle

Жоғалған нота

🎵 «Жоғалған нота» есебіне қош келдіңіз!

Біреу скрипкада әдемі C4 нотасын ойнап жатқанын елестетіңіз... Бірақ содан кейін біз дыбыстың негізгі жиілігін жоясақ ше? Пуф! ✨ Жоғалды!

Дегенмен, миыңыз әлі де C4 нотасын естиді. Ал жасанды интеллект осы музыкалық нотаны анықтай ала ма?

Бұл тапсырмада әрбір аудиоклип - дыбыстың иллюзиясы: шынайы нота жоқ және оның тек спектрлік дыбыстары ғана берілген. Әрбір мысалда негізгі жиілік (f_0) және бірнеше кездейсоқ гармониктер жойылды.

Осыған қарамастан, адам миы қалған гармоникалардың аралықтары мен құрылымын талдай отырып, дыбыстың дұрыс биіктігін қалпына келтіре алады.

🎯 Сіздің міндетіңіз: осы «толық емес» дыбыстан бастапқы музыкалық нотаны (мысалы, C4, G#5) жіктейтін модель жасау.

Сіз жасанды интеллектіні адам сияқты тыңдауға үйрете аласыз ба? 🎵🎵🎵

Сипаттама

Бұл аудиосигналдарды бірнеше классқа жіктеу туралы есеп.

Кіріс деректер:

Әрбір мысал дискреттеу жиілігі 22050 Гц болатын 16-бит PCM WAV форматындағы монофониялық аудиосигналдан тұрады. Әрбір файлдың ұзақтығы әртүрлі, бірақ әдетте 1-ден 4 секундқа дейін. Аудио файл алдын ала өңделді және кейбір жиіліктер жойылған.

Есептің мақсаты:

Әрбір аудиофайл үшін дыбыс биіктігін белгілеудің ғылыми жүйесінде көрсетілген бастапқы дыбыс биіктігін білдіретін музыкалық нотаны табу керек (мысалы, C4, F#5, Bb3). Белгілер жиынтығы 61 жеке класстан тұрады және әрбір сынып A0 (MIDI 21) -ден C8 (MIDI 108) -ге дейінгі MIDI ноталара сәйкес келеді.

Өзгертілген аудиосигнал арқылы (кейбір жиіліктерді алып тастау арқылы) бастапқы нотаны болжау керек. Модель негізгі жиіліктің немесе гармониктердің спектралдық және уақытша құрылымынан тон биіктігін (нотаны) шығаруы тиіс.

Бағалау

Сіздің шешіміңіз стандартты дәлдікті есептеу арқылы бағаланады:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of examples}}$$

🎯 Баға неғұрлым жоғары болса, шешіміңіз соғұрлым жақсы!

Файлдар

Тестілік жиынтықтағы әрбір дыбыс жолы үшін ноталар реңкі биіктігінің шифрланған белгісін болжау қажет. Жіберілетін **CSV** файлдың құрылымы төмендегідей болуы тиіс:

Path, Pitch_ID
path/sample1.wav, 0
path/sample2.wav, 1
path/sample3.wav, 2
etc.

Russian - Missing Fundamental Puzzle

Потерянная нота

🎵 Добро пожаловать в головоломку «Потерянная основа»!

Представьте, что кто-то играет красивую ноту C4 на скрипке... Но затем мы удаляем фактическую высоту звука. Пуф! ✂ Исчезла!

И всё же каким-то образом ваш мозг всё ещё слышит C4. Итак, вот главный вопрос: могут ли машины услышать музыкальную ноту, которой нет?

В этом задании каждый аудиоклип — это иллюзия высоты звука: истинная нота отсутствует, и остаются только её спектральные отголоски. В каждом сэмпле была удалена основная частота (f_0) и несколько случайных гармоник.

Несмотря на это, человеческий мозг всё ещё может восстановить правильную высоту звука, анализируя интервалы и структуру оставшихся гармоник.

🎯 Ваша задача: построить модель, которая классифицирует исходную музыкальную ноту (например, C4, G#5) из этого «неполного» аудио, имитируя способность мозга восстанавливать высоту звука по гармонической структуре.

Думаете, сможете научить машину слушать, как человек? Тогда ныряйте — и услышите неслышимое! 🎵🎵🎵

Описание

Это задача многоклассовой классификации аудиосигналов с одной меткой.

Входные данные:

Каждый сэмпл представляет собой монофонический аудиосигнал в формате 16-бит PCM WAV с частотой дискретизации 22050 Гц. Длительность каждого файла варьируется, но обычно составляет от 1 до 4 секунд. Аудиофайл был предварительно обработан, и из него были удалены некоторые частоты.

Цель:

Для каждого аудиофайла целью является отдельная метка музыкальной ноты, представляющая исходную высоту звука, выраженную в научной системе обозначений высоты звука (например, C4, F#5, Bb3). Набор меток включает 61 отдельных классов, где каждый класс соответствует одному из нот MIDI от A0 (MIDI 21) до C8 (MIDI 108), которые встречаются в данных.

Вам надо по изменённому аудиосигналу (с удалением некоторых частот) предсказать истинную исходную ноту, которая была воспроизведена изначально. Модель должна выводить высоту тона из спектральной и временной структуры оставшихся гармоник, без доступа к основной частоте или другим удаленным гармоникам.

Оценка

Представленные материалы оцениваются с использованием стандартной точности классификации:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of examples}}$$

🎯 Чем выше оценка, тем лучше ваши рекомендации!

Файлы

Для каждой аудиопути в тестовом наборе необходимо предсказать зашифрованную метку высоты тона нот. Файл должен содержать заголовок и иметь следующий формат:

```
Path, Pitch_ID  
path/sample1.wav, 0  
path/sample2.wav, 1  
path/sample3.wav, 2  
etc.
```