Технология для моделирования и анализа систем классификации на основе машинного обучения

Алимов Н. А.

Научный руководитель: д.ф.-м.н.
профессор Граничин О. Н.

Рецензент: Ерофеева В. А.

2017

Введение

- Развитие методов и подходов машинного обучения и анализа данных
- «Зоопарк» технологий
- Необходима высокая компетенция исследователя
- Необходимость каждый раз писать много кода (получение первого приемлемого решения)

Платформы для анализа данных и машинного обучения

- Amazon Machine Learning (компания Amazon)
- Azure Machine Learning (компания Microsoft)
- Google Cloud Machine Learning (компания Google)
- Другие

Существующие решения

	Проприетарная лицензия	Программная расширяемость	Быстрый старт	Baseline	Простота установки и развёртывания
Amazon Machine Learning	+	+	-	-	+
Azure Machine Learning	+	+	-	-	-
Google Cloud Machine Learning	+	-	-	-	-

Постановка задачи

Цель работы — разработать технологию, позволяющую эффективно решать задачи классификации и кластеризации. Для достижения этой цели были сформулированы следующие подзадачи.

Подзадачи

- Разработать архитектуру технологии
- Программно реализовать платформу, согласно предложенной архитектуре
- Провести тестирование реализованного продукта
- Продемонстрировать работу системы на основе трёх задач

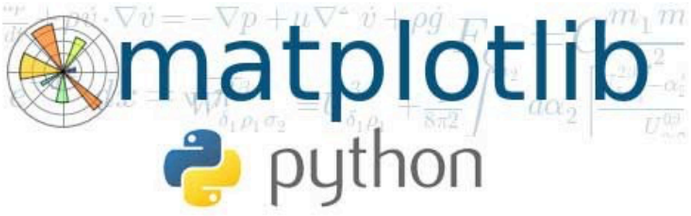
Требования

- Возможность загружать данные
- Добавлять свои алгоритмы
- Использовать существующие алгоритмы
- Быстрый старт

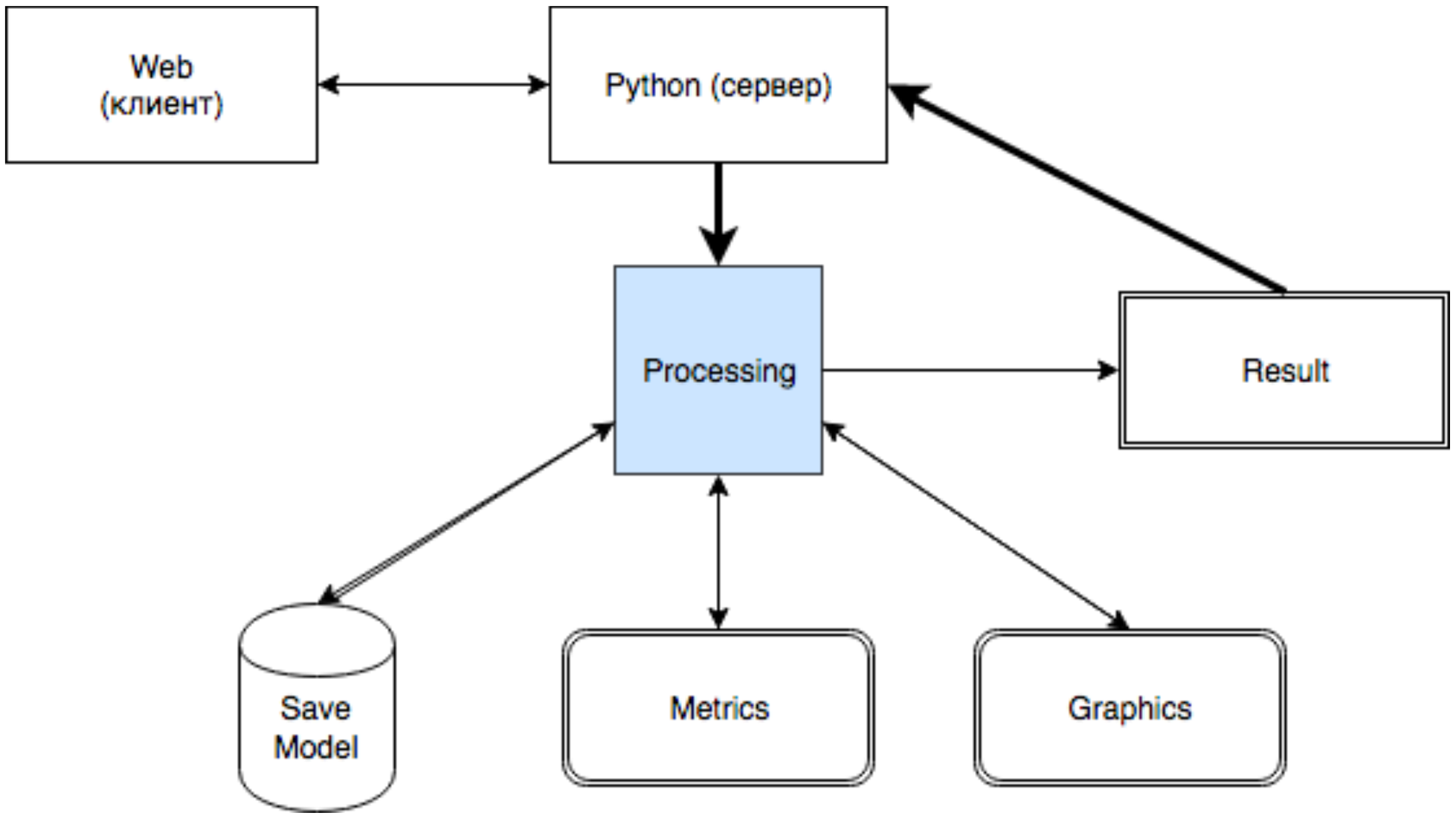
Технология

- Модель хранения данных
- Взаимодействие клиента и сервера
- Машинное обучение и анализ данных (processing part)
- Веб-интерфейс

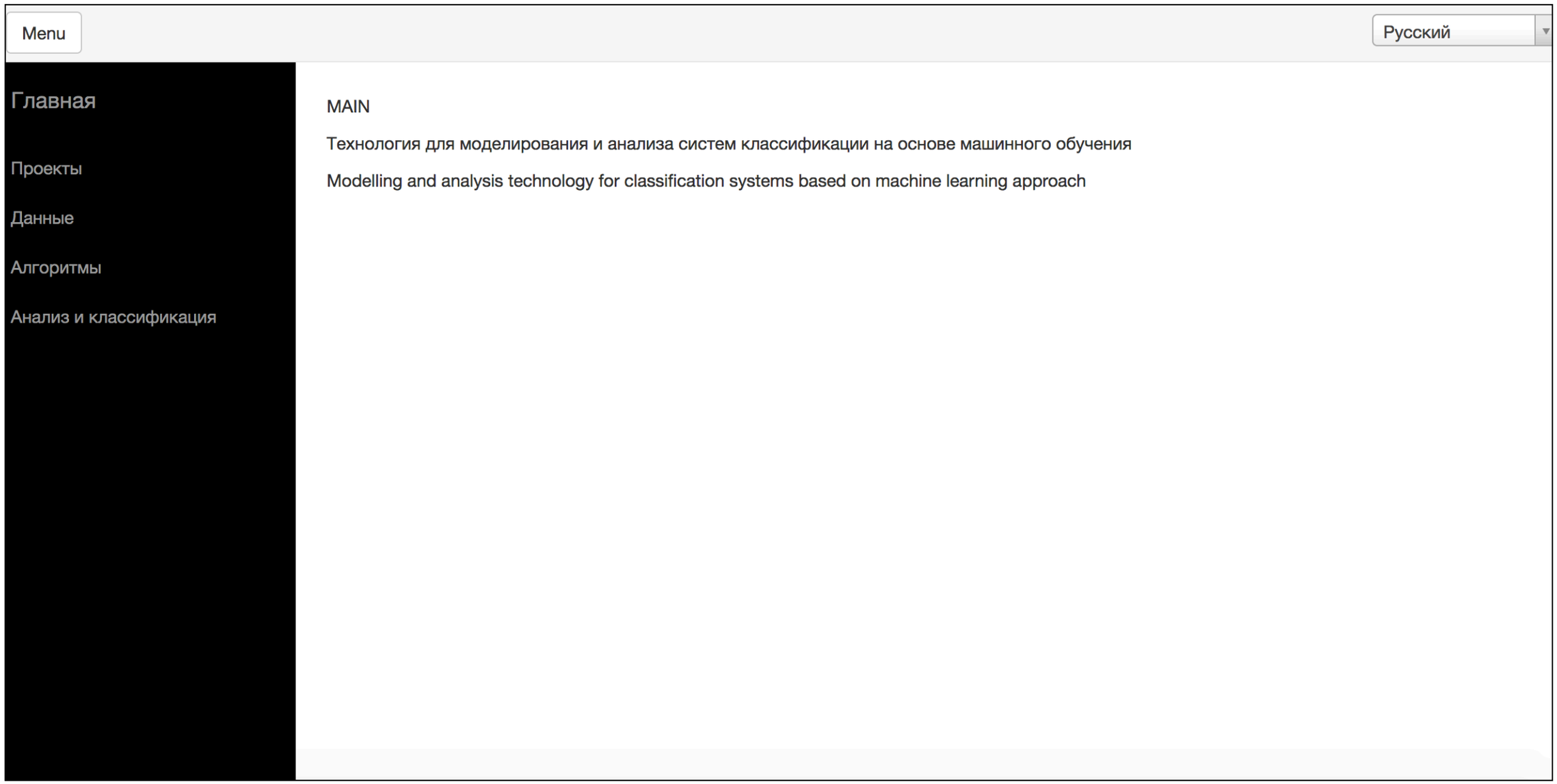
Программная реализация



Компоненты


Web
(клиент)
Python (сервер)
Processing
Result
Save
Model
Metrics
Graphics

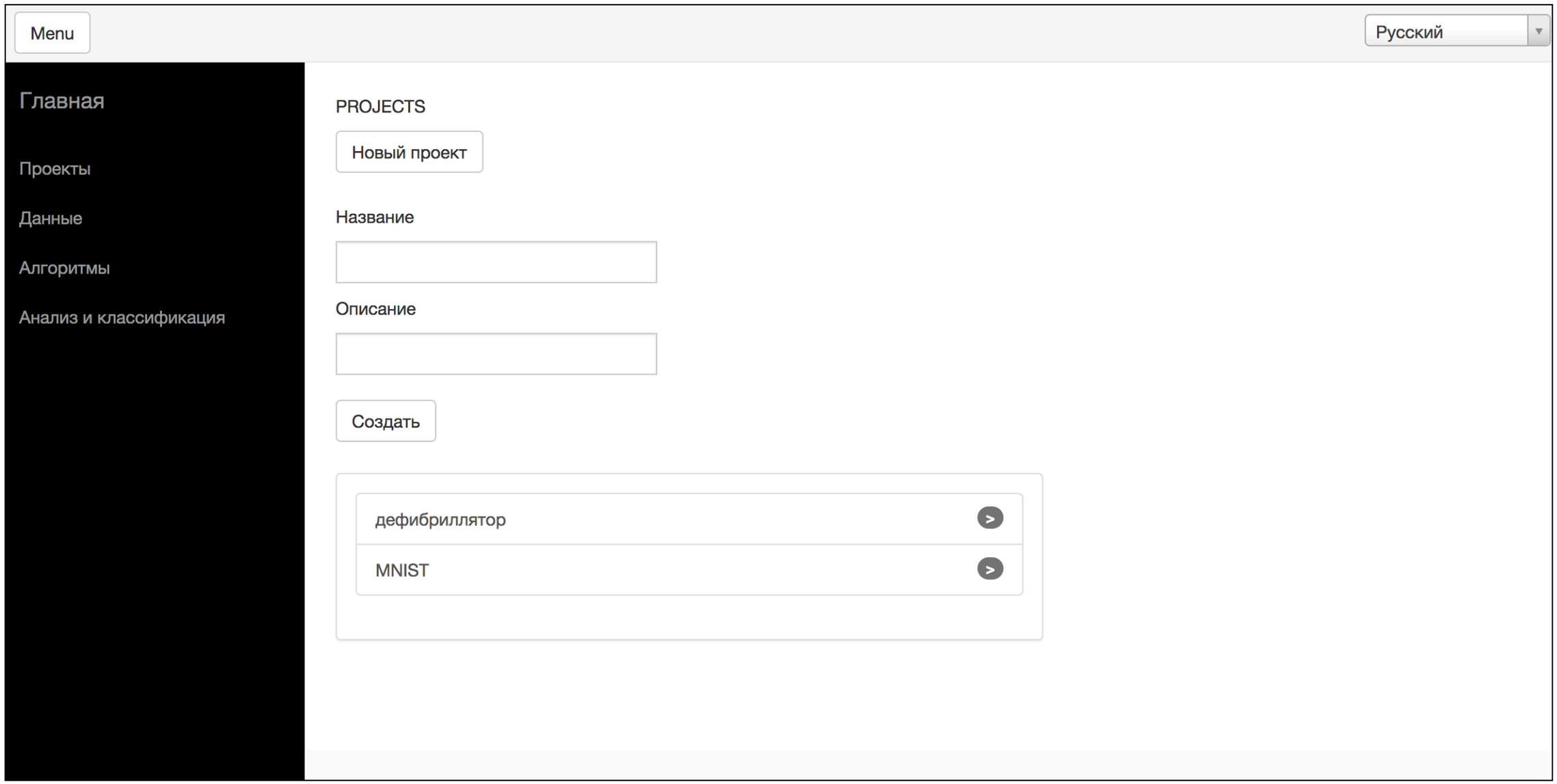
Главная страница



Проекты

- Возможность создать несколько проектов
- Проект - главная сущность
- Проект (название, описание, дата создания)

Проекты

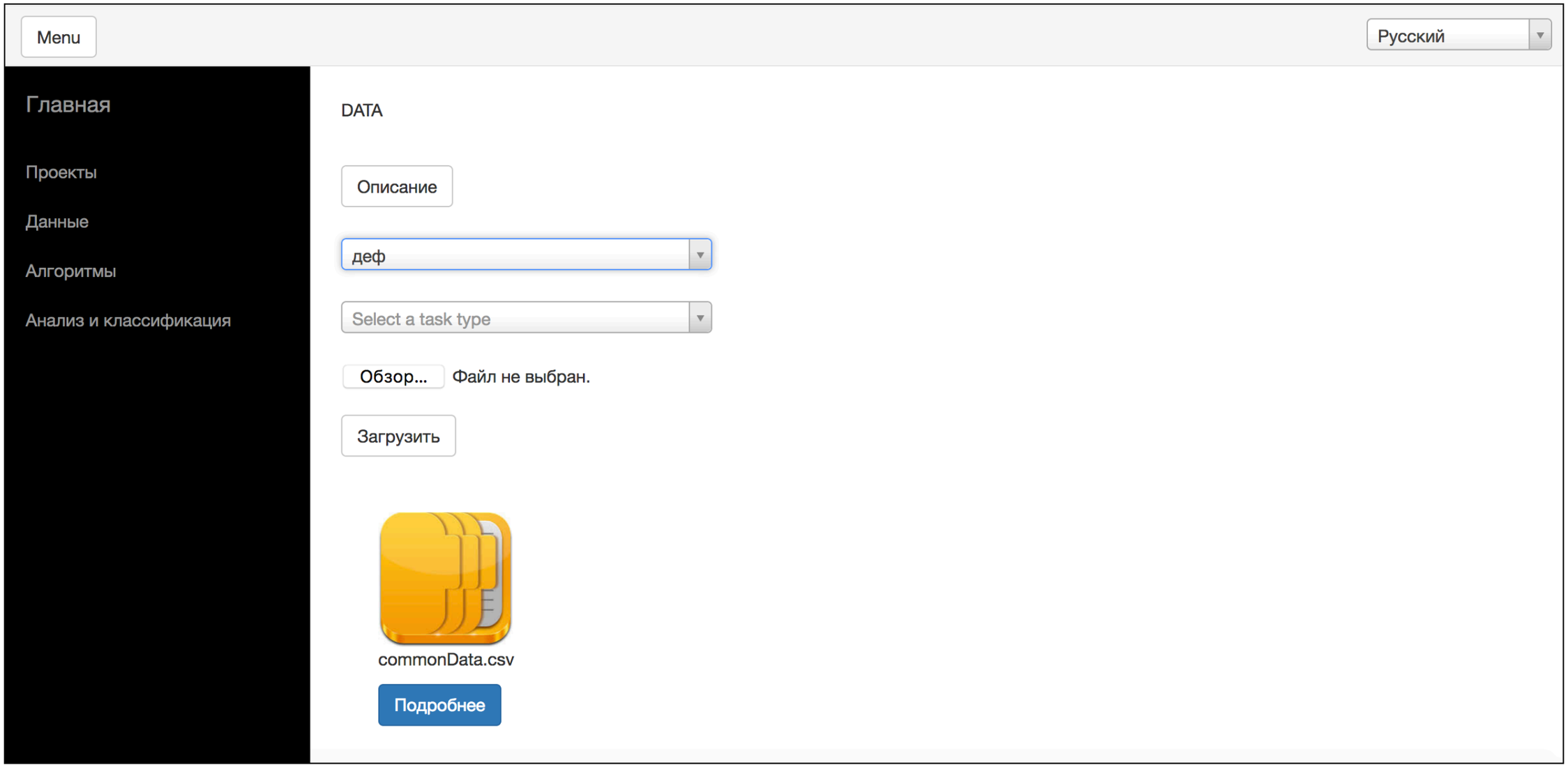

Menu
Русский
Главная
Проекты
Данные
Алгоритмы
Анализ и классификация
PROJECTS
Новый проект
Название
Описание
Создать
дефибриллятор
MNIST

Данные

- Формат данных должен быть единый (формат описания данных дан в примере)

- Поддерживаются (CSV, TSV)

Данные


Menu
Русский
Главная
Проекты
Данные
Алгоритмы
Анализ и классификация
DATA
Описание
деф
Select a task type
Обзор...
Файл не выбран.
Загрузить
commonData.csv
Подробнее

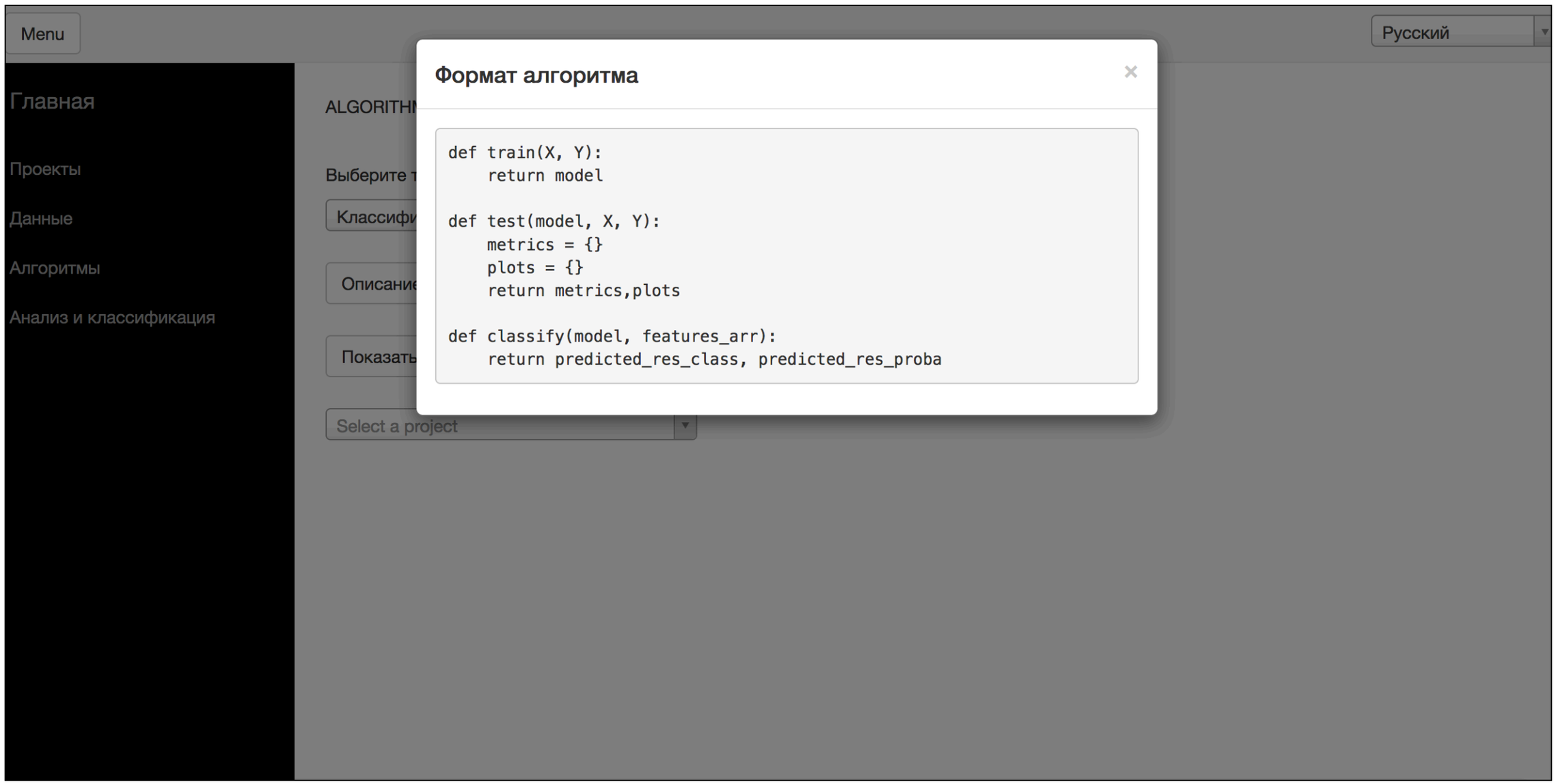
Алгоритм

- Некоторые алгоритмы уже предзагружены в с систему.

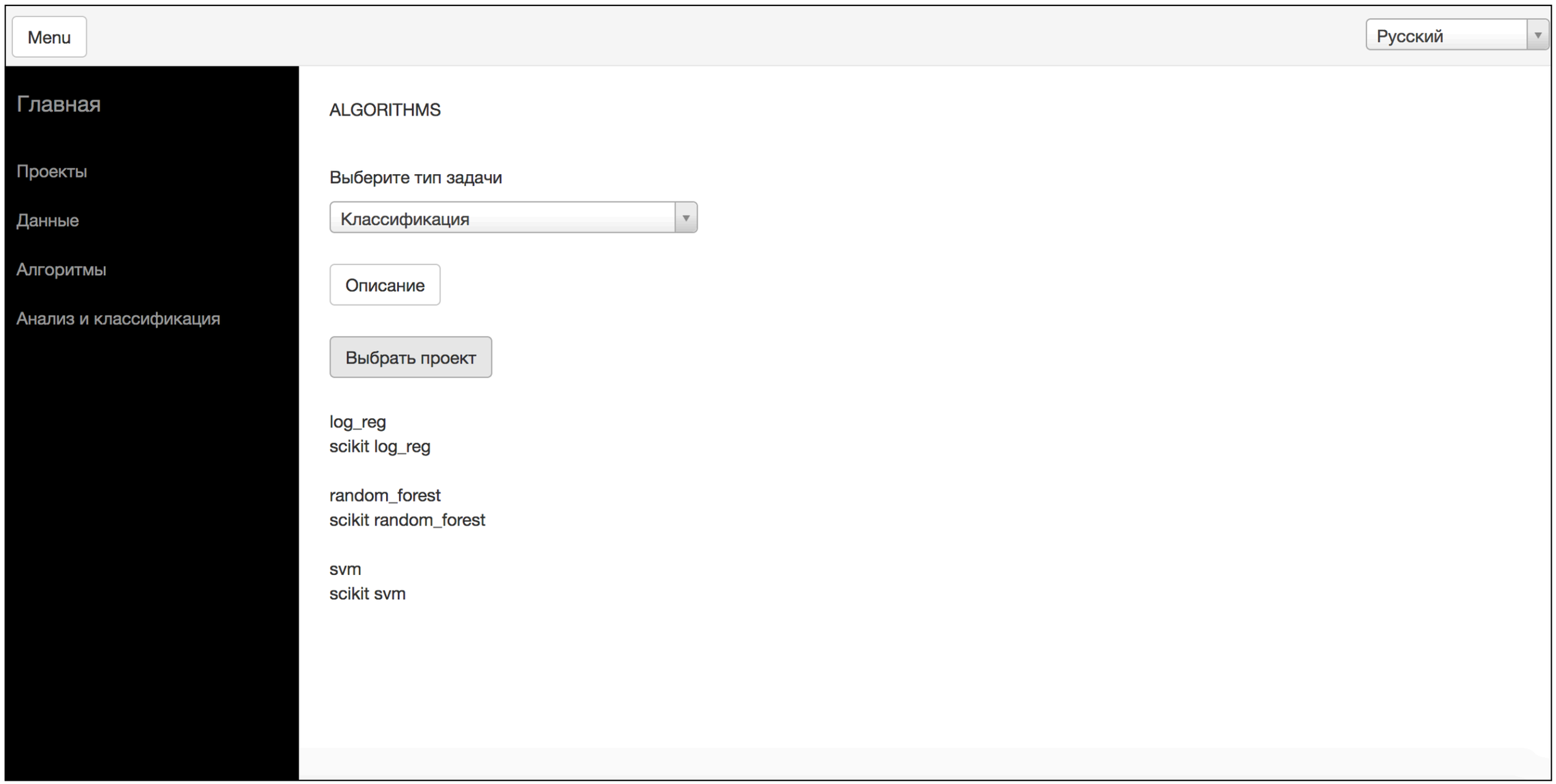
- Можно загрузить свои (в определенном формате).

- Поддерживается язык Python и библиотеки из его окружения.

Алгоритмы


Menu
Русский
Главная
Проекты
Данные
Алгоритмы
Анализ и классификация
ALGORITH
Выберите т
Классифи
Описание
Показать
Select a project
Формат алгоритма
def train(X, Y):
return model
def test(model, X, Y):
metrics = {}
plots = {}
return metrics,plots
def classify(model, features_arr):
return predicted_res_class, predicted_res_proba

Встроенные алгоритмы



Скрипт алгоритма готового к загрузке в систему

```python
import numpy as np
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics.classification import hamming_loss

from app.init_server import num_cpu


def train(X, Y):
    rf = RandomForestClassifier(n_jobs=num_cpu)
    rf.fit(X, Y)

    return rf


def test(model, X, Y):
    metrics = {}

    predicted_proba = model.predict(np.array(X))

    metrics['hamming loss'] = hamming_loss(Y, predicted_proba)

    plots = None

    return metrics, plots


def classify(model, features_arr):
    res_arr_class = [model.predict(np.array(i).reshape(1, -1)) for i in features_arr]
    res_arr_proba = [model.predict_proba(np.array(i).reshape(1, -1)) for i in features_arr]
    return res_arr_class, res_arr_proba

```

Анализ и классификация

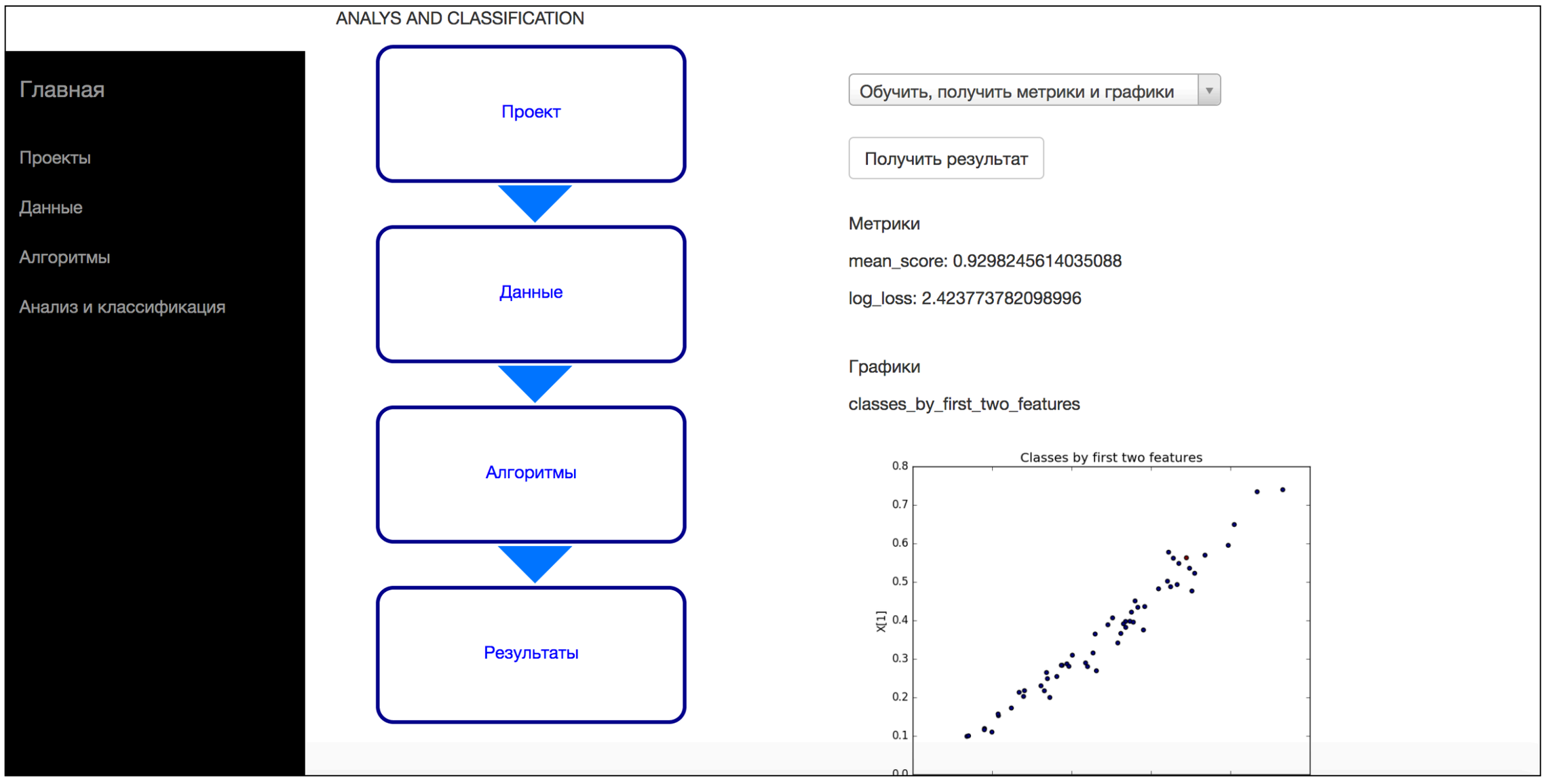
- Все проекты, алгоритмы, результаты и данные сохраняются в отдельную папку

- Алгоритмы, которые уже были обучены заново не обучаются (отпечаток данных и текстового кода алгоритма)

- Считаются метрики (предзагруженные и собственные)

- Строятся графики

Тестирование



Эксперименты

- Задача предсказания успешности дефибрилляции
- Кластеризация картинок с цифрами
- Классификация вин

Алгоритмы

Классификация

Стандартные (предзагружены в систему)

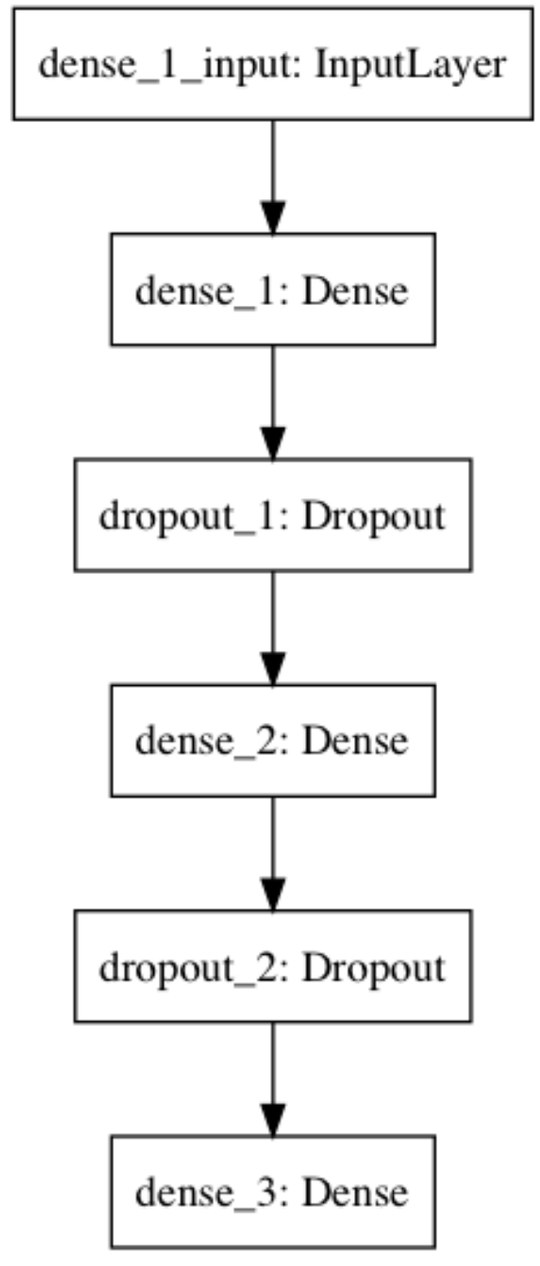
- Random Forest
- Logistic regression
- SVM

Собственный

- Нейронная сеть (keras)

Perceptron

- Оптимизатор: adam (https://arxiv.org/abs/1412.6980)
- Активация (сигмоид)
- 2 внутренних слоя (15, 10)
- dropout после каждого скрытого слоя (0.2, 0.1)
- loss: categorical_crossentropy
- 200 epoch, 5 per batch



Алгоритмы

Кластеризация

Стандартные (предзагружены в систему)

- k-means
- Birch
- Mini Batch KMeans

Метрики

Классификация

mean accuracy

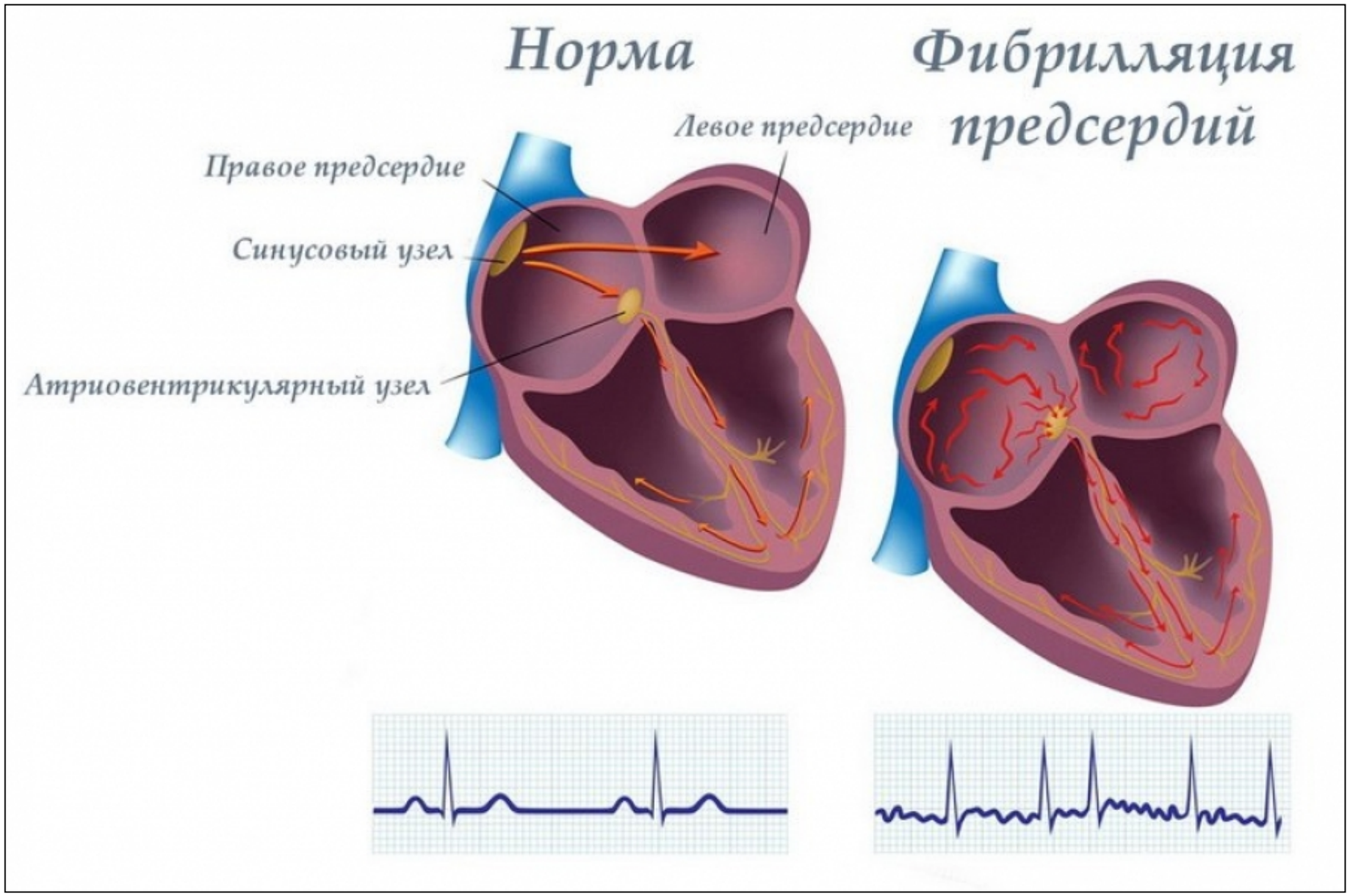
Кластеризация

adjusted rand score

Задача предсказания успешности дефибрилляции

- Получение данных в режиме реального времени
- Устройство, обрабатывающее данные и дающее рекомендации по проведению дефибрилляции
- Предсказание успешности лечения

Фибрилляция предсердий



Автоматический дефибриллятор

LifePak 12

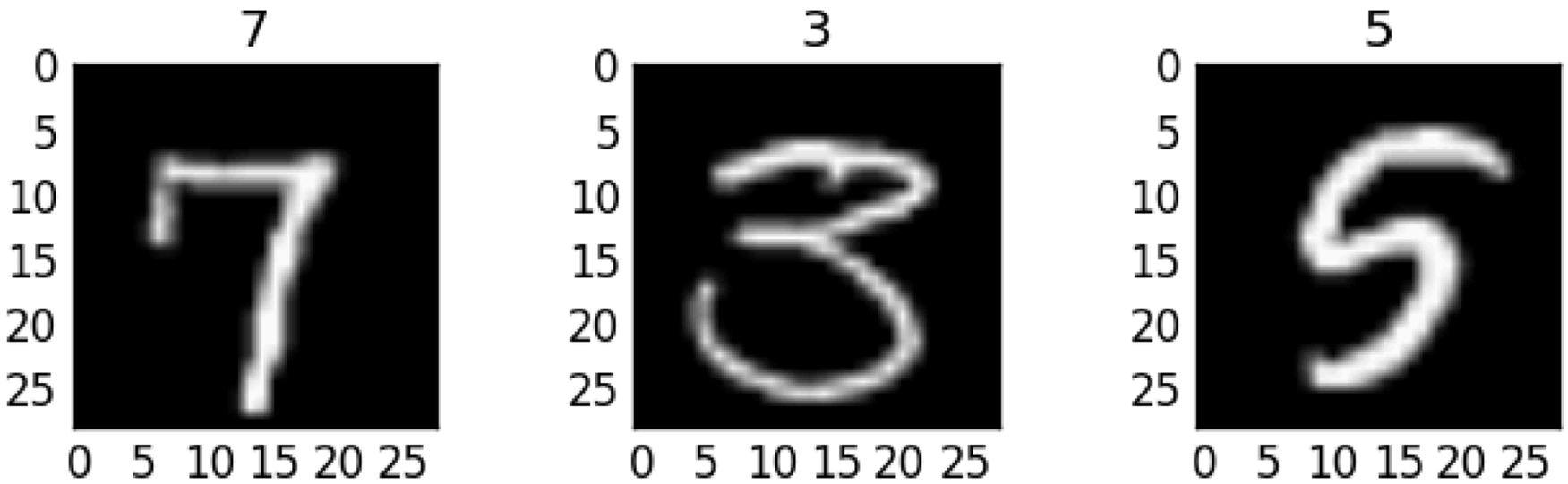
Сравнение алгоритмов

	mean accuracy
Random Forest	0.86
SVM	0.91
Logistic Regression	0.93
Perceptron	0.91
Random Algorithm	0.45

Задача кластеризации картинок с цифрами

- Задача состоит в том, чтобы на основе данных о картинках с цифрами провести их кластеризацию.

Данные



Кластеризация картинок с цифрами

	adjusted rand score
k-means	0.38
Mini Batch KMeans	0.37
Birch	0.30

Задача классификации вин

- Задача заключается в том, чтобы на основе химического анализа вина предсказать его сорт (один из трёх).

- Данные представляют собой набор векторов, размерностью 13. Всего 178 примеров. Всего три класса вин.

- Количество по каждому из сортов:
- class 1: 59
- class 2: 71
- class 3: 48

Классификация вин

	mean accuracy
Random Forest	0.95
SVM	0.42
Logistic Regression	0.98
Perceptron	0.74
Random Algorithm	0.34

Open source

- https://github.com/nsmalimov/diplom_ml_platform
- https://mlanalysisws.com
- необходимые пакеты / инструкция по развёртыванию
- простота установки, доработки

Результаты

- Разработана архитектура платформы
- Выполнена программная реализация
- Система протестирована.
- Работа системы продемонстрирована, с её помощью были решены несколько задач и проведён сравнительный анализ полученных решений

Список литературы

- Граничин О. Н. Поляк Б. Т. Рандомизированные алгоритмы оценивания и оптимизации при почти произвольных помехах. — М.: Наука, 2003. — P. 191.
- Granichin O. Volkovich Z. Toledano-Kitai D. Randomized Algorithms in Automatic Control and Data Mining. — USA, New York, 2015. — P. 275.
- Vapnik V. Chervonenkis A. Leader–follower consensus problems of multi-agent systems with noise perturbation and time delays // Soviet Math. Doklady. — 1968. — no. 9. — P. 915–918.
- Фомин В. Н. Математическая теории обучаемых опознающих систем. — Л.: Изд-во Ленингр. ун-та, 1976. — P. 236.

QA