

УДК 004.056

## ТЕХНОЛОГИИ ИНТЕГРАЦИИ МАШИННОГО ОБУЧЕНИЯ С ВЕБ ПРИЛОЖЕНИЯМИ

Сабитов<sup>1</sup> Б.Р., Сейтказиева<sup>2</sup> Н.С., Кубанычбекова<sup>3</sup> А.К.

<sup>1</sup>Кыргызский национальный университет им.Ж.Баласагына

<sup>2</sup>Кыргызский государственный университет им.И.Арабаева

<sup>3</sup>Кыргызский государственный университет строительства, транспорта и архитектуры им.Н.Исанова

Изучаются методы интеграции результатов моделирования машинного обучения с веб приложениями Python. Для конкретной модели определения болезней по симптомам из медицины строится веб приложение с помощью Фреймворка Django.

**Ключевые слова.** Прогнозирование, машинное обучение, модели, интеграция, веб приложение, Django.

## WEB ТИРКЕМЕЛЕР МЕНЕН МАШИНА ОКУУНУН ИНТЕГРАЦИЯЛАШУУ ТЕХНОЛОГИЯСЫ

Сабитов<sup>1</sup>Б.Р., Сейитказиева<sup>2</sup> Н.С., Кубанычбекова<sup>3</sup> А.К.

<sup>1</sup>Ж.Баласагын атындагы Кыргыз улуттук университети

<sup>2</sup>И.Арабаев атындагы Кыргыз мамлекеттик университети

<sup>3</sup>Н.Исанов атындагы Кыргыз мамлекеттик курулуш, транспорт жана архитектура университети

Машина үйрөнүү симуляциясынын натыйжаларын Python веб тиркемелери менен кантип интеграциялоону үйрөнүңүз. Дарыдан келген симптомдор боюнча ооруларды аныктоонун конкреттүү модели үчүн Django Framework аркылуу веб-тиркеме курулууда.

**Баштапкы сөздөр.** Болжолдоо, машина үйрөнүү, моделдер, интеграция, веб тиркеме, Django.

# TECHNOLOGIES OF INTEGRATION OF MACHINE LEARNING WITH WEB APPLICATIONS

**Sabitov<sup>1</sup>B.R., Seitkaziya<sup>2</sup> N.S., Kubanychbekova<sup>3</sup> A.K.**

<sup>1</sup>Kyrgyz National University named after Zh. Balasagyn

<sup>2</sup>Kyrgyz State University named after I. Arabaev

<sup>3</sup>N. Isanov Kyrgyz State University of Construction, Transport and Architecture

Learn how to integrate machine learning simulation results with Python web applications. For a specific model for determining diseases by symptoms from medicine, a web application is being built using the Django Framework.

**Keywords.** Prediction, machine learning, models, integration, web application, Django.

## ВВЕДЕНИЕ

В данной статье изучается, один из современных методов исследования прикладных задач, разработка веб приложение на Фреймворке Python с учетом результатов построения моделей машинного обучения. Данная направление является интеграция и развертывания результатов машинного обучения как веб приложение для конечных пользователей, которая в настоящее время является актуальной задачей.

Используя систему проектирования на Фреймворке Django в работе построена веб приложение с использованием сервера PostGreSQL для хранения базы данных. Изучается процесс обучения моделей в медицине и ее построение с применением алгоритмов машинного обучения –методов ближайших соседей и случайный лес, которые очень хорошо подходят к построению нелинейных моделей многих прикладных задач. Рассмотрен прикладная задача из медицины, что является определить болезни пациентов по симптомам. При построении моделей с применением мощных алгоритмов машинного обучения как случайный лес и ее применение для моделирования определения болезней пациентов по симптомам. Изучалось база данных, в которой отражены различные болезни и заболевания и по базе симптомов и модель учитывает и прогнозирует процесс протекания болезней. Для удобства использования модели в практических целях, в

диссертационной работе изучалось процесс интеграции машинного обучения с Интернет технологией , в частности с созданием веб портала в виде веб приложения с использованием продвинутой системы построения Интернет приложений Фреймворка Django.Удобства данного веб Фреймворка для построения многих сложных веб приложений является , что он имеет встроенный сервер для тестирования данных и собственный административный сайт, как веб сервер и который позволяет связаться с сервером базы данных. В качестве сервера базы данных использовано , промышленный сервер PostGreSQL, с помощью которого и веб приложения формируются данные базы данных. Изучено особенности процесса построения веб приложения с технологией Django и машинного обучения. Для построения модели использовано программная среда Jupyter системы Anaconda. Программный код реализован с использованием Python технологий. Мощный системный язык Python со многими библиотеками и интеллектуальными системами позволяет с применением программной среды Jupyter модели самой сложной структуры. Для создания веб системы использовано виртуальная среда, построенная на дисковом пространстве компьютера, для локализации задачи и используемых пакетов программ Python.

**Методы и результаты исследования.** Схема построения модели и веб системы для данной задачи приведен на Рис.1.



Рис.1. Схема построения модели и веб приложения для медицинских учреждений.

Для построения веб приложения в качестве сервера базы данных выбрано PostgreSQL, а в роли сервера приложений использовано встроенный административный сайт Фреймворка Django с сервером SQLite. Структура проекта содержит, аккаунты администратора, докторов и пациентов. Общий вид проекта на Django:

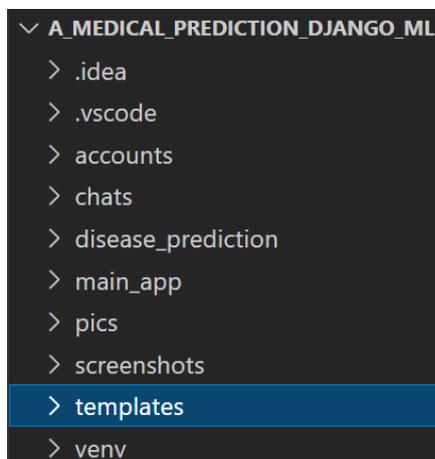


Рис.2. Структура проекта на Django для медицинских учреждений.

где, accounts приложение аутентификации, chats-чат связи пациентов с докторами, disease\_prediction-проект для прогнозирования, main\_app-основное приложение для определения прогноза болезни пациентов. Для построения веб приложения использовано технологии Django. Вот элементы главной страницы веб системы



Для администрирования базы данных в веб системе встроена аутентификация администратора, докторов и пациентов.

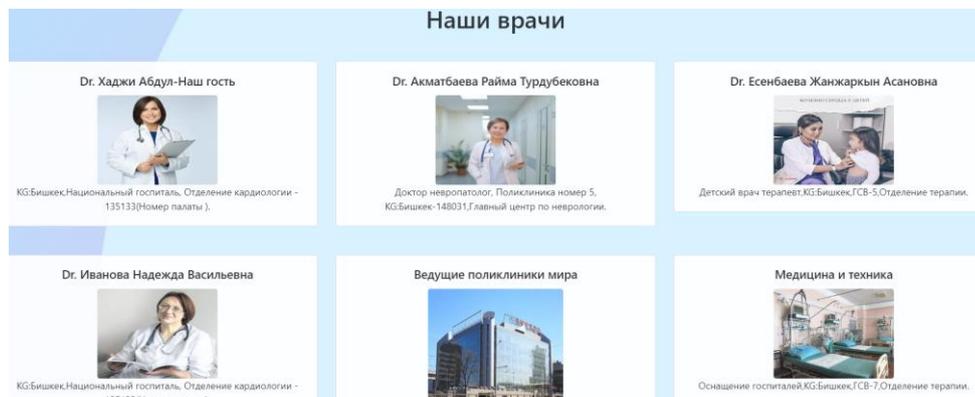
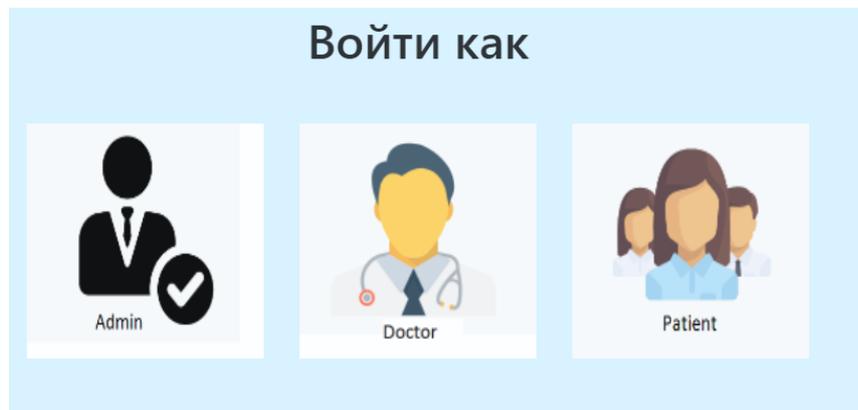


Рис.3. Главная страница веб системы.

Структура веб сервера на базе сервера приложений SQLite 3, встроенного в Django выгладить следующим образом:

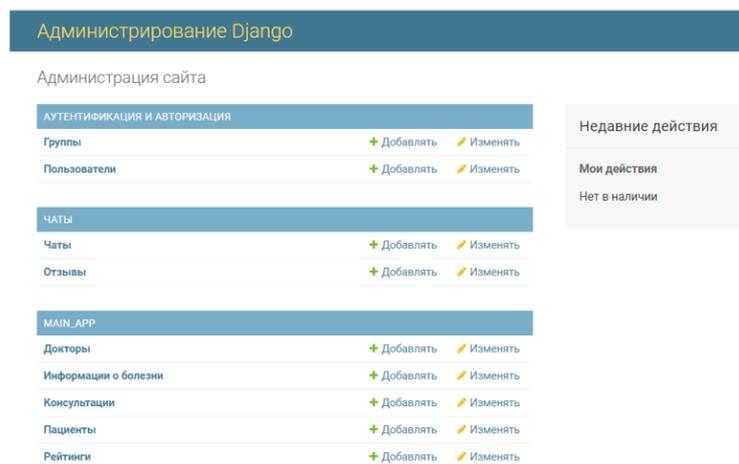


Рис.4. Структура встроенного административного веб сайта на Django.

Для администрирования базы данных созданы следующие модели: модели аутентификации и авторизации –группы и пользователи, модели чаты –чаты и отзывы и основной сайт, который состоит из моделей –доктора,

информации о болезни, консультация, пациенты и рейтинги докторов. Как видно из рис.4. в административном сайте для редактирования данных моделей встроена вкладки редактирования. Например, форма редактирования, для приложения Main\_app выглядит в виде формы

Администрация Main\_App

| ОСНОВНОЕ_ПРИЛОЖЕНИЕ  |  |
|----------------------|--|
| Докторы              | + Добавлять  Изменять |
| Информации о болезни | + Добавлять  Изменять |
| Консультации         | + Добавлять  Изменять |
| Пациенты             | + Добавлять  Изменять |
| Рейтинги             | + Добавлять  Изменять |

Рис.5.Форма редактирования для приложений.

Для редактирования отдельного элемента модели встроена форма редактирования, показано в следующей форме.

Изменить доктора

Пользователь: Асанов@mail.ru  +

Терпеливый

Доктор

Имя: АСАН

Доб: 2021-03-31 Сегодня   
Примечание. Вы опережаете серверное время на 5 часов.

Адрес: Бишкек

Номер мобильного: 05555555

Пол: муж

Номер регистрации: 12345

Год регистрации: 2022-02-15 Сегодня   
Примечание. Вы опережаете серверное время на 5 часов.

Квалификация: Доктор высшей квалиф

Государственный медицинский совет: Медицина

Специализация: Терапевт

Рейтинг: 12

Рис.6. Форма редактирования и изменения записей модели доктора в виде формы.

Форма базы данных симптомов и соответствующий связь с болезнями.

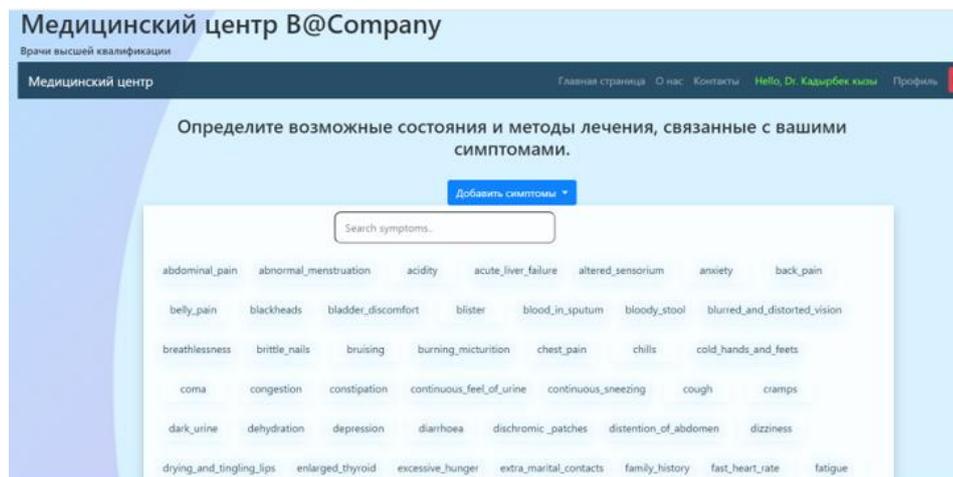


Рис.7. Форма для ввода симптомов болезни пациентов.

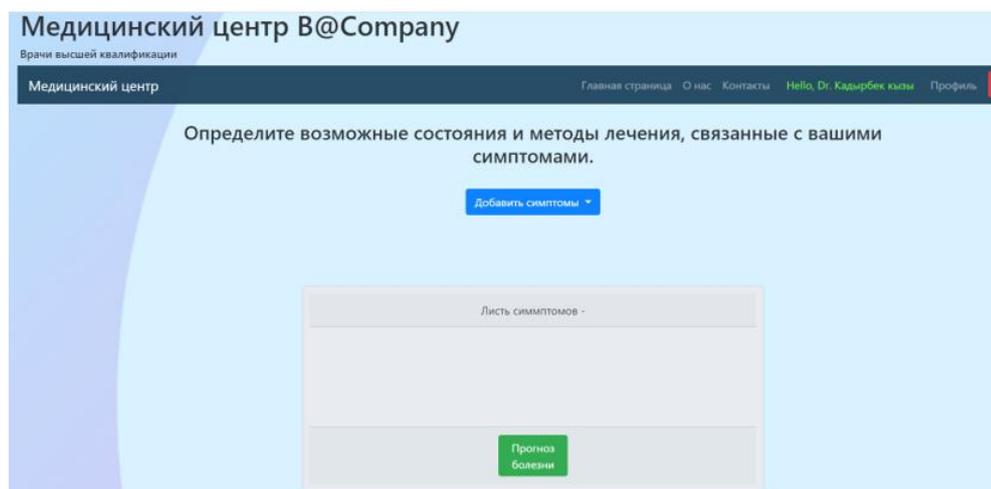


Рис.8. Форма для прогнозирования болезни.

Ниже будет представлен пациенту окончательный прогноз болезни пациента и в случае необходимости консультация с врачом. Для прогнозирования использовано предварительно обученный модель определения болезни по базе данных симптомов, которая сериализовано на диске компьютера. Для хранения симптомов использовано Java Script технологии. Форма диагноза с результатом представлен ниже (без указания данных и болезни пациента).

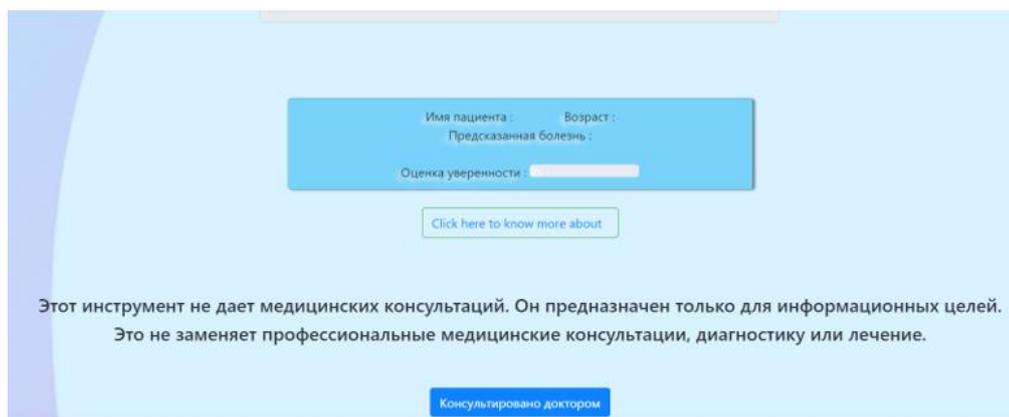


Рис.9.Окончательный прогноз болезни пациентов.

Данное приложение является учебно- познавательным приложением и не является, приложением по определению болезней без медицинских заключений и анализа дополнительных заключений врачей и соответствующих органов по практической реализации данного проекта.

**Заключение.** Изучено и разработано веб приложение с учетом результатов построения моделей с применением машинного обучения. На основе алгоритмов машинного обучения и веб приложение для практического использования конечными пользователями. Используя систему проектирования Django в работе построено веб приложение с использованием сервера PostGreSQL для формирования базы данных симптомов, болезней, докторов и пациентов. Обучение моделей осуществлялся с помощью мощных алгоритмов машинного алгоритма ближайших соседей и случайный лес, которые очень хорошо подходят к построению нелинейных моделей многих прикладных задач. Методы исследования в данной статье с успехом можно перенести к задачам определения болезни растений, животных и птиц.

## ЛИТЕРАТУРА

1. Дж. Грас – Data Science. Наука о данных с нуля, Мир,2020
2. О'Нил, Шатт – Data Science. Инсайдерская информация для новичков,Мир,2020

3. П. Брюс, Э. Брюс – Практическая статистика для специалистов Data Science,2020.
4. О'Нил, Шатт – Data Science. Инсайдерская информация для новичков,2020
5. Ын, Су – Теоретический минимум по Big Data. Всё что нужно знать о больших данных,2020.
6. Силен, Мейсман, Али – Основы Data Science и Big Data. Python и наука о данных,2020.
7. Дж. Вандер Плас – Python для сложных задач. Наука о данных и машинное обучение,2020