**Цель проекта по SMART**

| **Критерий** | **Цель** |
| --- | --- |
| Конкретный | Разработать рекомендательную систему для выбора образовательных учреждений на основе предпочтений пользователя и географических параметров. |
| Измеримый | Создать алгоритм, который будет выдавать 20 ранжированных результатов в зависимости от введённого города или региона пользователя.  Ранжированный список из топ-20 результатов ВУЗов, факультетов и специальностей с наибольшим количеством анкет с аналогичными значениями. |
| Достижимый | Проанализировать и обработать данные с помощью технических средств и алгоритмов обработки больших объёмов данных.  Разработать и протестировать функциональную систему рекомендаций с учетом географических ограничений. |
| Значимый | Предоставить пользователям индивидуализированные рекомендации по образовательным направлениям в зависимости от их интересов и выбранного региона.  Способствовать улучшению процесса выбора образовательных учреждений и повышению качества образования. |
| Ограниченный по времени | Завершить разработку рекомендательной системы в течение 18 дней с учётом работы команды из 6 человек.  Провести тестирование и подготовить документацию к завершению проекта. |

Конкретные задачи для достижения цели проекта перечислены в плане действий (файл *План действий.docx*).

**Устав проекта**

| **Наименование проекта** | Разработка ML-решения рекомендательной системы для UniGoTo |
| --- | --- |
| **Планируемое время начала и окончания проекта** | Начало 28.11.2023  Окончание 15.12.2023 |
| **Оценка бюджета проект** | 0 руб. 0 коп. |
| **Место/сфера реализации** | Информационные технологии, рекомендательные системы, образование, профориентация. |
| **Заказчик проекта** | Стартап UniGoTo |
| **Другие/ключевые участники проекта** | Команда разработки (6 человек) “Las Teteras Desesperadas”:  Загальский Игорь - проект-менеджер, дата-инженер, дата-аналитик, ML-инженер  Санникова Юлия - дата-аналитик, ML-инженер  Майнгерт Владимир - дата-инженер  Аванесян Тачат - технический писатель  Мухачев Иван - технический писатель  Антропова Наталия - дата-аналитик, QA-инженер  Собственник проекта: Панферов Кирилл. |
| **Дата создания документа** | 29.11.2023 |

**Цели проекта**

| Реализовать самостоятельное ML-решение рекомендательной системы, способное сформировать персональные ранжированные списки направлений обучения в ВУЗах для каждого уникального запроса на основе данных, вводимых пользователем рекомендательной системы. Разработать документацию. |
| --- |

**Описание проекта**

| Проект направлен на создание самостоятельного ML-решения для UniGoTo - рекомендательного сервиса ВУЗов и специальностей на основе технологии ИИ - и разработку документации. Решение формирует персональные ранжированные списки направлений обучения в ВУЗах для каждого уникального запроса, состоящего из таких данных, как музыка, фильмы, книги, игры или хобби.  **Этапы реализации (план проекта)**  Инициация проекта  ● Встреча с заказчиком  ● Формирование проектной команды  ● Подготовка устава проекта  ● Определение необходимых ресурсов  Планирование  ● Разработка плана действий  ● Составление таймлайна проекта и реестра рисков  ● Подготовка черновика решения  Сбор и обработка данных  ● Сбор данных  ● Обработка данных  Создание рекомендательной системы  ● Разработка ML-решения рекомендательной системы согласно спецификациям и требованиям проекта  ● Реализация ранжирования результатов по различным параметрам  Интеграция и тестирование  ● Интеграция компонентов решения  ● Тестирование решения на совместимость и работоспособность  ● Доработка функционала  Разработка документации  ● Создание технической документации  ● Документирование кода  Завершение проекта  ● Подготовка отчёта о реализации проекта  ● Подведение итогов  Дополнительные задачи (при наличии времени)  ● Реализация решения в виде Docker-контейнера  ● Реализация взаимодействия через API |
| --- |

**Требования к проекту**

| 1. Сроки и бюджет:  ○ Проект должен быть завершен в течение 18 дней с установленным бюджетом в размере 0 рублей.  2. Команда и ресурсы:  ○ Необходимо обеспечить наличие 6 человек со следующими специальностями: проект-менеджер, дата-инженер, дата-аналитик, ML-инженер, тестировщик, технический писатель.  ○ Команда должна иметь доступ к необходимым инструментам и технологиям для разработки ML-решения рекомендательной системы, API сервису заказчика.  3. Функциональные требования:  В проекте должно быть реализовано:  ○ Ввод - несколько строк, вывод - ранжированный список.  ○ Вывод топ-20 результатов ВУЗов, факультетов и специальностей с наибольшим количеством анкет с аналогичными значениями в формате списка.  ○ Предусмотрена возможность выдачи результатов по ВУЗам, факультетам и специальностям по отдельности, а также с ограничением по городу или региону. Должно указываться в начале запроса до начала сравнения.  4. Технические требования:  ○ На основе больших данных, переданных заказчиком в виде API, произвести обработку данных (убрать пустые строки, шумы), а также разработать самостоятельную рекомендательную модель и документацию.  ○ Документация должна содержать информацию по обработке первоначальных данных, обучению, сборке, использованию и переобучению.  ○ Код должен быть хорошо структурирован, легко читаем и понятен. Должны быть включены смысловые комментарии и подходящие имена переменных и функций.  ○ Решение должно:  - быть работоспособно;  - закрывать бизнес-задачу;  - быть разработано командой.  ○ Более подробное описание содержится в техническом задании. |
| --- |

# **Техническое задание**

# **на разработку рекомендательной системы,**

# **для проекта UniGoTo**

**Цели рекомендательной системы**

Выдача персонального ранжированного списка направлений обучения в высших учебных заведениях на основе данных, вводимых пользователем рекомендательной системы.

**Условия выполнения**

На основе больших данных, переданных заказчиком в виде API, произведите обработку данных (убрать пустые строки, шумы), а также разработайте самостоятельную рекомендательную модель и документацию. Документация должна содержать информацию по обработке первоначальных данных, обучению, сборке, использованию и переобучению.

Плюсом будет реализация в виде Docker-контейнера и реализация взаимодействия в виде API, предпочтительнее в виде GRPC с описанием схемы взаимодействия.

**Предположительные этапы работ**

Сборка данных, обработка данных для реализации системы, создание рекомендательной системы, тестирование, написание документации.

**Описание API сервиса для получения входных данных**

API будет предоставляться по адресу unigoto.ru/api/input для доступа будет использоваться входные POST-параметры:

● token (обязательный) — данный параметр содержит токен авторизации, выдаётся индивидуально, строка ненулевой длины.

● page (обязательный) — содержит номер страницы выходных данных, строка не нулевой длины, состоящая из номера в диапазоне (1-700000), данные отсортированы от старых к новым. Также рекомендуем учесть, что объём данных достаточно большой и использовать его весь не имеет смысла — важен подход к решению задачи, его финальная реализация выходит за рамки Хакатона.

**Ограничения**

Лимит на количество запросов и количество этих запросов составляет 10 в минуту.

**Описание выходных данных**

Выходные данные отдаются после успешного запроса и имеют формат json.

Массив из 999 элементов имеет значения:

**about** (string) — содержимое поля «О себе»

**activities** (string) — содержимое поля «Деятельность»

**books** (string) — содержимое поля «Любимые книги»

**games** (string) — содержимое поля «Любимые игры»

**interests** (string) — содержимое поля «Интересы»

**country** (object) — информация о стране, указанной на странице пользователя в разделе «Контакты». Возвращаются следующие поля:

**id** (integer) — идентификатор страны

**title** (string) — название страны

**city** (object) — информация о городе, указанном на странице пользователя в разделе «Контакты». Возвращаются следующие поля:

**id** (integer) — идентификатор города

**title** (string) — название города

**education** (object) — информация о высшем учебном заведении пользователя. Возвращаются поля:

**university** (integer) — идентификатор университета

**university\_name** (string) — название университета

**faculty** (integer) — идентификатор факультета

**faculty\_name** (string)— название факультета

**graduation** (integer) — год окончания