**А.С. Чеботаревский[1], Е.А. Левчук[2]**
[1]УО «Гомельский государственный университет
имени Франциска Скорины», Гомель, Беларусь
[2]УО «Белорусский торгово-экономический университет
потребительской кооперации», Гомель, Беларусь

# ПОСТАНОВКА ЗАДАЧИ НА АВТОМАТИЗАЦИЮ ФОРМИРОВАНИЯ БЮДЖЕТА КОМАНДИРОВОЧНЫХ РАСХОДОВ

Главными статьями командировочных расходов являются расходы на транспорт и проживание в отеле. Причем среди транспортных расходов наиболее затратными являются авиасообщения. В данный момент формирование авиамаршрутов происходит в режиме поиска по сайтам авиакомпаний или посредников. Авторам не известны решения, которые позволили оптимизировать полный цикл командировочных расходов.

Частные решения работают с узким списком авиакомпаний и позволяют формировать маршруты не сложнее, чем маршрут из точки А в точку Б с использованием нескольких промежуточных точек на жестко заданные даты. Использование жестко заданных дат не всегда позволяет получить приемлемый результат.

Второй класс существующих решений – это логистические методы, осуществляющие поиск по замкнутому графу с помощью достаточно устаревших алгоритмов. Для поиска оптимальных маршрутов при использовании линейных алгоритмов специалисты сталкиваются с тем, что требуются большие вычислительные мощности для поиска оптимальных маршрутов.

Используя большой объем данных, которые предоставляют авиакомпании и сервисы по поиску отелей, можно составить множество моделей, которые позволят с определенной точностью предсказать изменение цен и обнаружить зависимости, которые можно будет использовать для составления будущих маршрутов на основе спрогнозированных изменений цен. Использование линейных алгоритмов для поиска по замкнутому графу не является оптимальным и требует большой объем вычислительных мощностей. Но при использовании нейронных сетей и построения LSTM сети можно сэкономить достаточно вычислительной мощности и получить более эластичные и интересные результаты. На данный момент не существует ни одного решения, которое бы позволило составить оптимальный маршрут с точки зрения стоимости и времени, учитывая большое количество вариантов и возможностей.

Таким образом, задача разработки сервиса, который позволит автоматизировать процесс формирования бюджета командировочных расходов является оптимальной.

Данная задача включает в себя три глобальные подзадачи:

1 поиск и нормализация данных,

2 обработка нормализованных данных,

3 построение итоговых маршрутов и поиск оптимальных маршрутов.

Для решения первой подзадачи (поиск и нормализация данных), будут использоваться три вида поиска информации. Первый - поиск и работа с открытыми API, которые позволяют осуществить выборку необходимых данных. Большинство сервисов, которые являются агрегаторами авиакомпаний или отелей, предоставляют только закрытый API, стоимость которого зависит в большинстве случаев от количества запросов к их API. Так как требуется постоянный доступ к API, итоговая стоимость вполне может стать слишком большой при условии отсутствия инвестирования.

Второй способ более изящный. Для того, чтобы претворить его в жизнь, требуется использовать веб-сервер NodeJS в связке с GooglePuppeter, который является “headless chromium browser”. Использование данной технологии вместо классического парсинга и краулинга обязательно, так как все агрегаторы используют множество механизмов защиты от потокового парсинга информации, которая находится на их серверах. В случае с использованием GooglePuppeter становится возможным произвести полную эмуляции человеческого поведения на подобных сайтах. В результате клиент получает доступ

к информации в автоматизированном режиме, как будто данный парсинг производится в ручном режиме.

Третий способ парсинга является классическим вариантом, так как он заключается в “глупом” парсинге. С помощью Python и плагинов для веб краулинга есть возможность перехватывать некоторые виды данных, которые передаются от сервера к клиенту и обрабатывать их. В случаях со многими сайтами авиакомпаний клиент обрабатывает веб сокеты, что также можно использовать, перехватывая эти потоки данных. Наконец, нельзя забывать о дешифровке данных, полученных с помощью третьего способа. Обычно используются простейшие методы шифрования, которые обходятся без каких-либо проблем.

Следующая часть данной подзадачи является наиболее простой, так как требуется нормализовать полученные данные. Фактически это приведение данных к одному формату. Для того, чтобы не тратить время на построение миграций и структур базы данных, будет использоваться NoSQL СУБД, такая как MongoDB. Она позволит хранить изначально различные структуры данных и позже при помощи сериализаторов привести их к общему виду.

Вторая подзадача (обработка нормализованных данных) является наиболее творческой. Ее также можно декомпозировать на следующие подэтапы: построение веб-сервиса, решение задачи построения маршрутов, входящей в определение MVP.

Построение веб-сервиса - это наименее затруднительная часть проекта и на данный момент она частично реализована. В качестве технологий по реализации веб-сервиса со стороны backend используются следующие технологии: NodeJS, MongoDB, Mongoose, Express, PassportJWT, Multer, GraphQL, Apollo. Для frontend используются следующие технологии: VueJS, Apollo, Axios. Внешний вид строится с помощью css фреймворка Bootstrap4.

Решение задачи, входящей в определение MVP (minimum viable product), - это авторский способ классификации задачи. Его суть заключается в следующем. При разработке большого проекта в первую очередь следует сконцентрироваться на тех задачах, которые в условиях коммерческой ценности данного проекта, позволят осуществить запуск MVP, который будет иметь пользовательскую ценность. В случае с разработкой веб-сервиса по решению задачи построения оптимальных маршрутов, минимально необходимой частью проекта будет построение маршрутов, но в отличие от финальной версии проекта на данном этапе происходит полный отказ от всех технологий и методологий, которые могут увеличить временные расходы на запуск

MVP. Итого, при разработке данной подзадачи все внимание уходит на создание линейных “глупых” алгоритмов поиска и построения маршрутов. По сути в данном случае происходит отказ от использования нейронных сетей, предикативных механик и имитационного моделирования в пользу быстрых в контексте реализации линейных алгоритмов.

Третья подзадача (построение итоговых маршрутов) включает в себя: построение математических моделей, постановка экспериментов на построение маршрутов, создание рекуррентных нейронных сетей и их обучение, создание предикативных механик. Так как решение данной подзадачи напрямую зависит от выполнение подзадачи один и минимального решения подзадачи два, то на данный момент сложно говорить о точном планировании выполнения данной подзадачи. Самое главное, стоит иметь в виду последовательность внедрения технологий. Решение третьей подзадачи будет проходить методом последовательного внедрения функционала и последовательного увеличения сложности. Можно начать с построения маршрутов между двумя точками, продолжить построением плавающих маршрутов с использование различных методик, которые позволяют экономить. Например, методика “город-призрак”, где сравниваются прямые и пересадочные маршруты. Во втором случае, как это ни странно, можно получить более дешевый способ добраться до точки назначения.

Таким образом, при использовании нейронных сетей и большой обучающей выборке возможно найти закономерности, которые позволяют построить модели прогнозирования цен, а также оптимизировать будущие маршруты, не имея данных для будущих маршрутов.