МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

МЕХАНИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ

Кафедра дифференциальных уравнений и системного анализа

Интеллектуальный анализ статистических данных поездок такси

Курсовая работа

Бельской Екатерины
Артуровны

студентки 2 курса,
специальность 1-31 03 09
Компьютерная математика
и системный анализ

Научный руководитель:
кандидат физ.-мат. наук,
доцент Л. Л. Голубева

Минск, 2021

ОГЛАВЛЕНИЕ

Введение

Увеличение количества информации характерно практически для каждой сферы общественной деятельности. Для эффективной обработки и последующего применения полученной информации используются различные методы математического и статистического анализа.

Интеллектуальный анализ данных представляет собой процесс обнаружения необходимых сведений в крупных массивах данных. Его преимущественное использование заключается в следующем. При традиционном осмотре данных из-за чрезмерного объема информации, а также достаточно сложных связей между отдельными компонентами некоторые признаки и закономерности нельзя обнаружить.

В интеллектуальном анализе можно выделить два этапа: аналитический и описательный. Описательный этап включает себя представление требуемых для работы данных в удобном графическом виде: столбчатые и линейные диаграммы, гистограммы. Аналитический этап включает в себя использование следующих методов:

1. Статистическое наблюдение – сбор данных в соответствии с выбранными характеристиками.
2. Сводка данных и определение метода выборки – создание группировок, разделение всего массива на основании каких-либо признаков.
3. Получение новых признаков.

Целью представленной курсовой работы является изучение различных методов работы с данными, использование методов интеллектуального анализа, а также подготовка исходного массива данных для предсказания стоимости поездки путем машинного обучения.Исследование производится на основе статистических данных поездок желтого такси Нью-Йорка за 2020 год, представленных комиссией по такси и лимузинам города Нью-Йорка (TLC). Данные о поездках предоставлены лицензированными водителями транспортных средств, а также поставщиками технологических услуг (TSP), обеспечивающих такси электронными счетчиками.

Глава 1 Чтение данных

1.1 Общая структура статистических данных

Данные представляют собой набор CSV-файлов. CSV (от англ. Comma-Separated Values) — текстовый формат, предназначенный для представления табличных данных. Каждая строка таблицы соответствует строке текста, которая содержит одно или несколько полей, разделенных запятыми.

Название признака	Описание
VendorID	Код, указывающий поставщика TPEP, предоставившего запись. 1 = Creative Mobile Technologies, LLC; 2 = VeriFone Inc.
Trep_pickup_datetime	Дата и время, когда был задействован счетчик.
Trep_dropoff_datetime	Дата и время, когда счетчик был отключен.
Passenger_count	Количество пассажиров в транспортном средстве. Вводится водителем.
Trip_distance	Пройденное расстояние в милях, о котором сообщил таксиметр.
PULocationID	Зона такси, в которой таксиметр был включен.
DOLocationID	Зона такси, в которой был отключен таксиметр.

RateCodeID	Окончательный тарифный код, действующий в конце путешествия. 1= Standard rate 2=JFK 3=Newark 4=Nassau or Westchester 5=Negotiated fare 6=Group ride
Store_and_fwd_flag	Эта характеристика указывает, хранилась ли запись о поездке в памяти транспортного средства перед отправкой, так называемая «stored and forward», потому что транспортное средство не имело соединения с сервером. Y= хранимая и пересылаемая запись поездки N= не хранимая и не пересылаемая запись поездки
Payment_type	Цифровой код, указывающий на то, как пассажир оплатил поездку. 1= Кредитная карта 2= Наличные деньги 3= Бесплатно 4= Договорная стоимость 5= Неизвестно 6= Аннулированная поездка
Fare_amount	Стоимость проезда по времени и расстоянию, рассчитанная по счетчику.
Extra	Различные дополнительные расходы и надбавки. В настоящее время сюда входят только сборы за час пик и ночь в размере $0,50 и $1.
MTA_tax	$0.50 MTA налог, который автоматически взимается в зависимости от используемой расчетной ставки.
Improvement_surcharge	$0,30 налог на «улучшение». Данный вид доплаты начал взиматься в 2015 году.

Tip_amount	Сумма чаевых - это поле автоматически заполняется для чаевых по кредитной карте. Чаевые наличными в это поле не включены.
Tolls_amount	Общая сумма всех оплаченных в поездке сборов.
Total_amount	Общая сумма, взимаемая с пассажиров. Не включает чаевые наличными.

Таблица 1.1 Описание признаков исходных данных

1.2 Различные методы считывания данных

Для обработки данных был выбран высокоуровневый язык программирования Python. Python предоставляет огромное количество библиотек для работы с большими данными. В процессе выполнения был проведен сравнительный анализ двух различных библиотек Vaex и Pandas.

Vaex. Vaex - это библиотека Python для Out-of-Core Data Frames (подобно Pandas), для визуализации и изучения больших табличных наборов данных. Она может вычислять статистические показатели, такие как среднее значение, сумма, общее количество, стандартное отклонение и т.д., на N-мерной сетке со скоростью до миллиарда объектов/строк в секунду. Визуализация осуществляется с помощью гистограмм, графиков плотности, что позволяет интерактивно исследовать большие данные. Vaex использует распределение памяти, политику копирования с нулевой памятью и ленивые вычисления для достижения наилучшей производительности (память не расходуется впустую).

Для исследования предложенного набора данных с помощью библиотеки Vaex необходимо было конвертировать файлы формата CSV в файлы формата HDF5, с которыми Vaex сможет работать.

Hierarchical Data Format, HDF (Иерархический формат данных) — название формата файлов, разработанного для хранения большого объема цифровой информации. HDF5 является современной версией вышеуказанного формата.

Структура данных такого формата организована подобно иерархической файловой системе, и для доступа к отдельным частям информации применяются файловые пути.

При анализе с помощью инструментов библиотеки Vaex самым затратным по времени и по памяти был процесс конвертации файлов: исходные файлы были изначально достаточно большими, после конвертации HDF5-файлы оказались ещё больше. Однако открытие и обработка таких файлов происходит мгновенно, поскольку в память ничего не копируется. Данные только отображаются в памяти, что позволяет считывать их исключительно при необходимости.

Как было сказано ранее, библиотека обеспечивает возможностью не только вычисления каких-либо статистических показателей, но и отображения того, что содержат данные. Например, ниже представленный график показывает связь между количеством поездок (по вертикали) и количеством пассажиров в транспортном средстве на момент совершения поездки (по горизонтали).

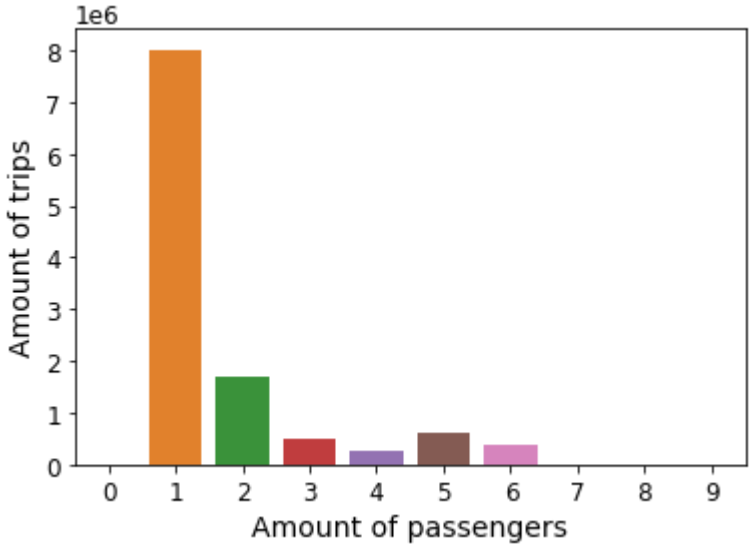

Рисунок 2.1 Столбчатая диаграмма (количество пассажиров на момент поездки)

Возможно представление и в несколько ином виде. Данная гистограмма отображает информации о количестве поездок различных по дальности поездки. По вертикали — количество поездок, по горизонтали — дальность.

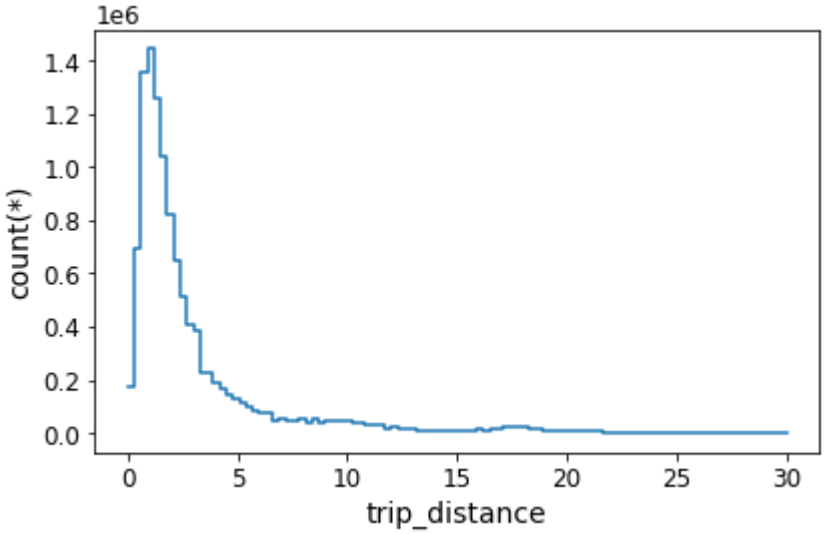

Рисунок 2.2 Гистограмма (дальность поездки)

Кроме того, используя plot, мы можем получить быстрый обзор того, откуда чаще всего совершались поездки (по данным 2015 года). Как видно из графика, места расположения пунктов отправления хорошо очерчивают Манхэттен.

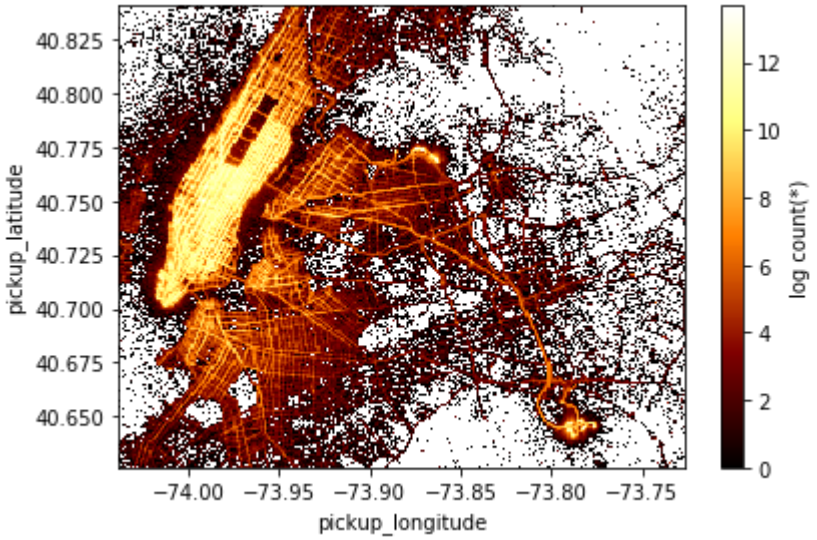

Рисунок 2.3 График пунктов отправления такси

Несмотря на быструю и выгодную по памяти обработку и визуализацию при дальнейшем выполнении заданий курсовой работы возникли некоторые проблемы. Так, при добавление новых столбцов, безошибочная обработка массива данных происходила в большинстве случаев только при первых нескольких проходах по таблице. После этого возникали ошибки, и необходимо было заново загружать данные в Data Frame и создавать по новой столбцы, что при проведении исследовательской работы является времязатратным. Аналогичная ситуация наблюдалась и при анализе срезов таблицы. По этой причине работа с данной библиотекой была прекращена.

Pandas. Pandas — ещё одна библиотека, используемая для обработки и анализа данных. Данная библиотека очень эффективна при работе с относительно небольшими объемами информации, и производительность редко становится проблемой. Pandas предоставляет структуры данных для анализа in-memory (в процессе обработки данные хранятся в памяти), что делает использование pandas для анализа наборов данных, превышающих объем памяти, несколько затруднительным. Даже те наборы данных, которые занимают лишь небольшую часть памяти, становятся громоздкими, поскольку некоторые операции pandas требуют создания промежуточных копий. При работе со статистическими данными поездок такси за один год такая проблема заполнения памяти иногда возникала даже не смотря на то, что данные сами по себе не являются слишком большими в определении больших данных в рамках используемой библиотеки.

Существует однако несколько способов оптимизации при работе с Pandas:

1. Считывание данных за один раз “порциями”, которые могут быть размещены в памяти. Такой способ является выгодным, например, в том случае, когда нужно вывести новый признак объекта на основе некоторых других, при этом финальное значение никак не зависит от показателей других объектов. При необходимости получения какого-либо статистического значения исходя из значений всех объектов таблицы процесс вычисления несколько затрудняется, так как за раз обрабатывается лишь какая-то часть всех необходимых данных.
2. Считывание лишь необходимых для работы столбцов таблицы.
3. Изменение типов данных столбцов. Типы данных pandas по умолчанию не являются наиболее эффективными с точки зрения использования памяти. Это особенно справедливо для столбцов текстовых данных с относительно небольшим количеством уникальных значений. Поэтому используя более эффективные типы данных, можно соответственно хранить в памяти более крупные массивы данных.

В дальнейшем выполнение заданий курсовой работы будет осуществляться при помощи инструментов именно этой библиотеки.

Глава 2 Предобработка данных

2.1 Объединение файлов

Так как данные поездок такси разделены на CSV-файлы в соответствии с месяцем их проведения, то для получения годовых характеристик необходимо было вычислить нужные показатели для данных каждого из двенадцати файлов. Существует два способа того, как это можно сделать: рассматривать каждый CSV-файл по отдельности и в конце объединять результаты или же объединить исходные файлы в один на начальном этапе. Преимущество первого метода заключается в том, что учитывая размер исходного одного файла проблем в нехваткой памяти не возникало бы. Однако в этом случае усложнилась бы структура кода. В данной работе был выбран второй способ, так как даже конечный файл с данными по поездкам желтого такси Нью-Йорка за 2020 год после объединения не слишком большой по размеру, поэтому его обработка не вызывает особых трудностей. Кроме того, на начальном этапе считывания CSV-файлов в DataFrame можно указать имена столбцов для считывания, тем самым избегая загрузки информации по ненужным для решения главной задачи признакам.

Для объединения двенадцати файлов в один была использованы библиотеки Glob и Pandas. Ранее не использованный модуль Glob находит все имена путей, соответствующие заданному шаблону, в соответствии с правилами, используемыми Unix, хотя результаты возвращаются в произвольном порядке. Путь может быть как точно прописан, так и указан относительно. Например, “D:/YT Data/*.csv” .

Таким образом находятся все соответствующие шаблону файлы с помощью метода glob (входные параметры: название пути) в указанной директории и заносятся в отдельную переменную. С помощью метода Pandas read_csv происходит чтение каждого найденного CSV-файла в DataFrame. После этого все полученные DataFrame конкатенируются.

Входными параметрами для метода read_csv являются filepath_or_buffer (путь к файлу), index_col (столбцы для использования в качестве меток), header (номера строк для использования в качестве названия столбцов DataFrame). Входные параметры для метода concat: objs(объекты для конкатенации), axis (ось для конкатенации), ignore_index (при значении True значения вдоль оси не используются для конкатенации).

2.2 Удаление лишних столбцов

Главная задача заключается в том, чтобы предобработать данные, найти в них новые признаки, а также подготовить эти данные для машинного обучения с целью предсказания стоимости поездки. Для достижения этой цели не все признаки исходных данных являются необходимыми.

Столбцы, которые можно удалить:

- Store_and_fwd_flag. Данный признак хранит в себе лишь информацию о том, как хранилась запись о проведенной поездке, поэтому не является необходимым.
- Payment_type. То, как пассажир оплатил поездку, также не влияет на конечную стоимость.
- Tip_amount. Размер чаевых не входит в стоимость поездке и основывается на желании пассажира.
- PULocationID и DOLocationID. Информация о зоне включения и выключения таксометра тоже является излишней в задаче предсказания стоимости.

Как было сказано раннее, удалить лишние столбцы можно на начальном этапе при выгрузке данных в DataFrame. Кроме того, это можно сделать с помощью метода drop, указывая названия столбцов для удаления или же индексы вдоль оси для удаления.

2.3 Удаление выбросов из таблицы

В анализе данных выброс — это какое-либо значение, которое значительно отличается от других значений в случайной выборке. Выброс может быть вызван изменчивостью измерений или может указывать на экспериментальную ошибку, последнее в большинстве случаев исключаются из набора данных. Так выбросы могут вызвать серьезные проблемы при статистическом анализе, поэтому их необходимо удалять.

Для удаления 1% выбросов для признаков trip_distance и fare_amount данные по этим столбцам были отсортированы и взяты срезы, исключая 0.05% минимальных значений и 0.05% максимальных. Срез — способ представления некоторой части последовательности. Получение среза аписывается в виде obj[start:stop:step], где start — начальный индекс, stop — конечный индекс, step — шаг выборки.

После этого однако построенный график плотности показал, что среди значений по данным характеристикам присутствуют отрицательные значения, которые также являются выбросами и были удалены аналогично при помощи срезов. График плотности — инструмент для визуализации распределения данных на каком-то интервале. Для построения графика плотности к объекту DataFrame была применена функция plot.kde без входных параметров.

2.4 Стандартизация данных

Стандартизация данных — это обеспечение внутренней целостности, то есть каждый тип данных имеет одинаковое содержание и формат. Стандартизированные значения полезны для отслеживания данных, которые нелегко сравнить иным способом. Для стандартизации пропущенных значений в признаках trip_distance и fare_amount были взяты срезы с условием для нахождения строк таблицы с такими значениями. Однако строк с пропусками в соответствующих признаках не было найдено.

2.5 Преобразование типов данных

Методом оптимизации при работе с большими массивами данных является преобразование типов данных. Типы данных pandas по умолчанию не являются наиболее эффективными с точки зрения использования памяти. Поэтому преобразование типов позволяет уменьшить объем хранимой в памяти информации. Например, по умолчанию все значения столбцов за исключением trep_pickup_datetime и trep_dropoff_datetime имеют тип float64, то есть для представления одного значения используется 64 бита. Однако принимая во внимание хранимые значения, такое выделение памяти является излишним. Кроме того, в таблице присутствуют значения, которые могут быть представимы в целочисленном виде (столбцы ‘VendorID’, ‘passenger_count’,’RatecodeID’).

Для преобразования типа объекта в указанный числовой тип используется функция to_numeric библиотеки pandas с входными параметрами arg (объект), downcast (параметр, указывающий на получаемый тип, а именно ‘float’ — для получения минимального типа float dtype (мин.: np.float32), ‘unsigned’— для получения минимального типа unsigned int dtype (min.: np.uint8)). В случае с преобразованием в целочисленный тип необходимо применение функции astype (приведение объекта pandas к указанному типу), а также функции fillna для заполнения значений NaN каким-либо не пустым значением (в работе использовалось нулевое значение).

Глава 3 Получение новых признаков

Для визуализации зависимости стоимости поездки, а также количества вызовов такси в зависимости от времени дня, дня недели и месяца необходимо создание новых временных признаков.

3.1 Преобразование типов

Применение функции dtypes (получение типа значений каждого столбца DataFrame) показало, что тип значений trep_pickup_datetime и trep_dropoff_datetime — object, а значит, для получения новых временных признаков нужно преобразовать значения в данных к столбцах к типу datetime. С этой целью используется функция to_datetime, входными параметрами которой являются arg (объект для конвертирования) и format (строка-формат преобразования, например, '%Y-%m-%d %H:%M:%S' для trep_pickup_datetime и trep_dropoff_datetime). Строки с пропущенными значениями в указанных столбцах были удалены.

3.2 Получение признаков

В качестве новых признаков были созданы:

- pickup_day и dropoff_day — название дня недели, когда таксометр был включен и после выключен. Используемая функция: dt.day_name().
- pickup_hour и dropoff_hour — час, когда таксометр был включен и выключен. Используемая функция: dt.hour.
- pickup_month — название месяца, когда таксометр был включен.

Данные признаки не являются важными для решения задача, однако помогают визуализировать зависимость количества поездок от различных временных промежутков.

Кроме вышеуказанных признаков для каждой поездки была найдена ее длительность в минутах trip_duration. Для этого из значений столбца tpep_dropoff_datetime вычитались значения столбца tpep_pickup_datetime, и полученная разность делилась на numpy.timedelta64(1, 'm'). После чего конечные значения были преобразованы к типу float (функция pandas.to_numeric(arg,

downcast=float")) Деление необходимо из-за того, что тип полученной разности — numpy.timedelta64.

NumPy — библиотека, предоставляющая поддержку больших многомерных массивов и матриц, а также большую коллекцию математических функций высокого уровня для работы с этими массивами.

3.3 Очистка полученных данных от выбросов

При анализе столбца trip_duration оказалось, что некоторые значения являются отрицательными или настолько большими, что исходя из них, поездка могла бы продолжаться несколько недель. Очевидно, что такие значения возникли из ввода ошибочных данных и являются выбросами, поэтому подлежат удалению. Удаление происходило, как и в Главе 2 п. 2.3, с помощью срезов с условием, а также сортировки.

Глава 4 Анализ данных

Анализ данных — это процесс изучения, оптимизации, преобразования и моделирования данных с целью обнаружения полезной информации, обоснования выводов и поддержки принятия решений. Важным инструментом анализа данных является визуализация.

4.1 Визуализация зависимости количества и стоимости поездок от времени их совершения

Для того, чтобы определить, когда люди чаще всего совершают поездки, построим соответствующие гистограммы. Для построения гистограмм воспользуемся библиотекой seaborn. Seaborn — это библиотека визуализации данных, основанная на Matplotlib. Она предоставляет высокоуровневый интерфейс для построения красивых и информативных статистических графиков. Matplotlib, в свою очередь, есть полнофункциональная библиотека для создания статических, анимированных и интерактивных визуализаций на языке Python.

Ниже представленные графики показывают, что в Нью-Йорке люди вызывают такси с субботы по понедельник включительно реже, чем в остальные

дни. Кроме того, можно увидеть, как влияет время суток на количество поездок. Самым нагруженным является промежуток времени с 13 часов дня до 19 часов вечера, и количество поездок, совершенных ранним утром, значительно меньше, чем количество, соответствующее поздней ночи.

Особый интерес представляет гистограмма распределения поездок по месяцам года. Из графика видно, что после января количество поездок сократилось приблизительно в 6 раз. По данным наиболее полного на сегодняшний день исследования о том, когда началось распространение вируса, искра, положившая начало эпидемии коронавируса в США, возникла в течение трех недель с середины января по начало февраля, до того, как страна приостановила поездки из Китая. Штат Нью-Йорк быстро стал эпицентром пандемии (Нью-Йорк имел наибольшее число подтвержденных случаев заболевания среди всех штатов с начала вспышки в США и до 22 июля), и был введен локдаун, о чем и свидетельствует показанная гистограмма распределения поездок такси. В гистограммах по вертикали — количество поездок, по горизонтали — временные промежутки.

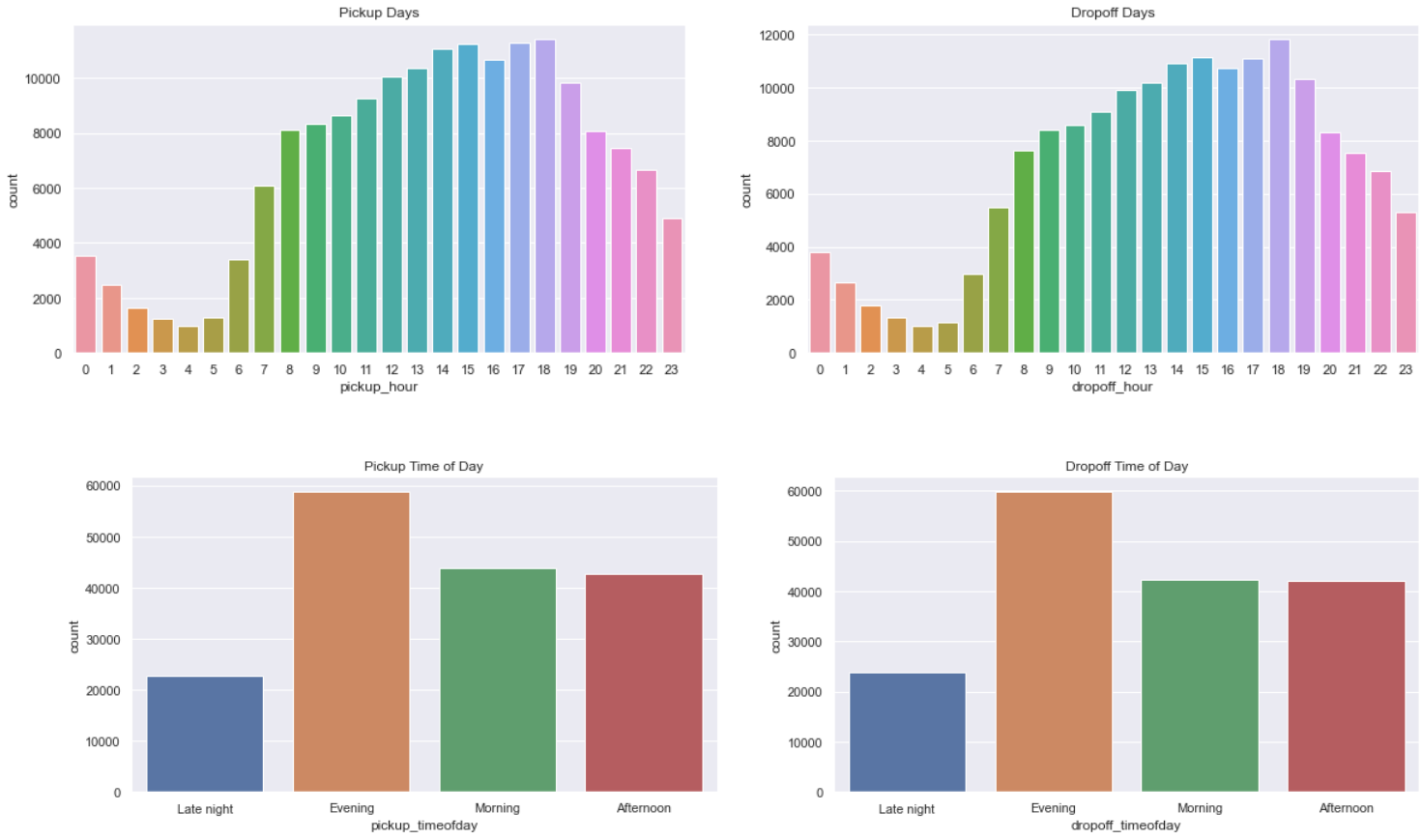

Рисунок 4.1 Гистограммы (количество поездок и время суток)

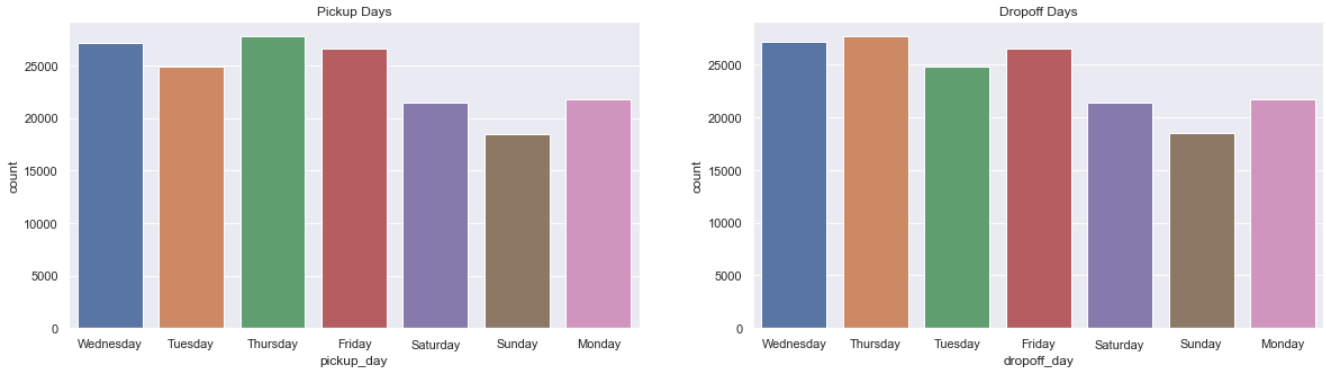

Рисунок 4.2 Гистограммы (количество поездок и дни недели)

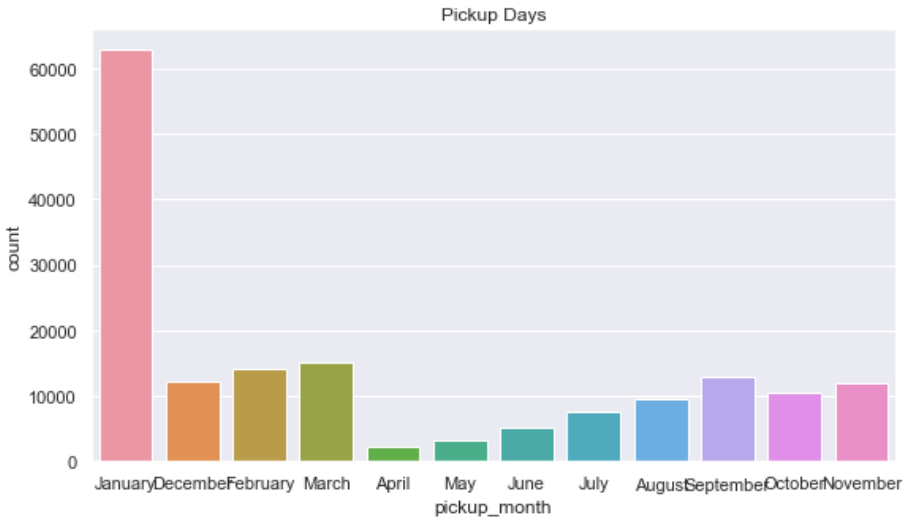

Рисунок 4.3 Гистограмма (количество поездок и месяцы года)

Для того, чтобы определить зависимость между стоимостью поездки и временем суток, построим график данной зависимости. Из представленного графика видно, что наиболее дорогие поездки совершаются в период поздней ночи, а также раннего утра. Максимальная стоимость достигается приблизительно в 4 часа 30 минут. Поездки, которые совершаются с 6-7 часов утра (час-пик) и до 14 часов дня являются наиболее дешевыми. Стоимость вечерней поездки также не сильно большая по отношению к ночному тарифу. На графике вдоль вертикальной оси расположены значения стоимости, вдоль горизонтальной — временные промежутки.

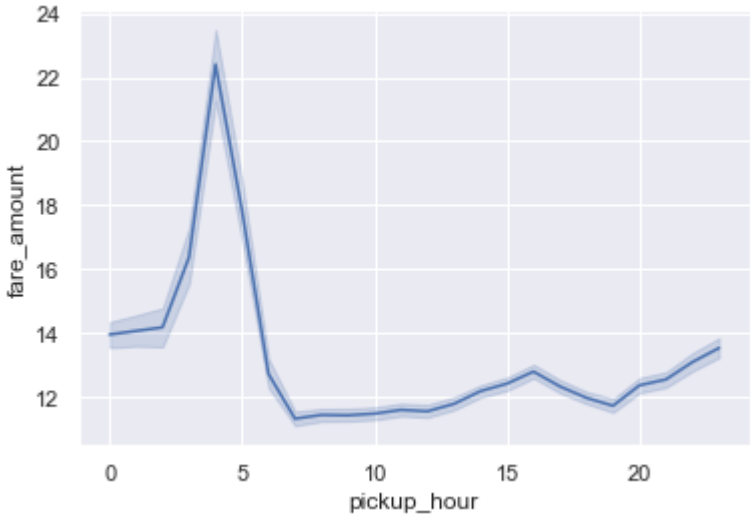

Рисунок 4.4 График зависимости стоимости поездки от времени ее совершения

4.2 Визуализация стоимости поездки от ее длительности

Построим график зависимости стоимости поездки от ее длительности и добавим цветовое разделение в зависимости от типа совершаемой поездки. По вертикали — значения стоимости, по горизонтали — продолжительность поездки.

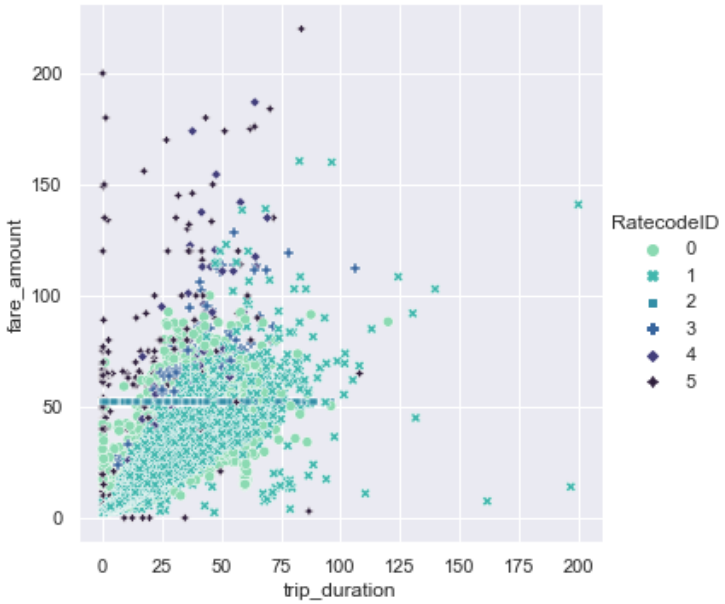

Рисунок 4.5 График зависимости стоимости поездки от длительности

Исходя из приведенного графика, какой-то строгой зависимости между данными признаками не наблюдается. Это связано с тем, что стоимость одной и той же по длительности поездки варьируется в пределах изменения времени суток. Однако наблюдается тенденция увеличения стоимости с увеличением времени в пути. Помимо этого, стоимость одного из типов поездок (JFK) неизменяема с течением времени. RatecodeID со значение 0 означает, что информация по типу поездки не была указана. RatecodeID со значением 6 не изображен на графике, так как поездок с таким типом очень мало.

4.3 Визуализация характеристик в пространстве

Введем новые характеристики — медианные значения стоимости, дальности и длительности поездки (‘fare_amount’, ‘trip_distance’, ‘trip_duration’). Для того, чтобы вычислить необходимые признаки в зависимости от типа поездки важными функциями являются groupby, принимающая на вход список столбцов для группировки, и функция median, вычисляющая непосредственно медианное значение. После этого визуализируем полученные характеристики в пространстве. Ось Ox будет обозначать медианное значение стоимости поездки, ось Oy — медианное значение дальности поездки, ось Oz — медианное значение длительности соответствующей поездки. На рисунке видно, что поездки JFK, Newark, Nassau or Westeches имеют наибольшие медианные значения стоимости, дальности и длительности.

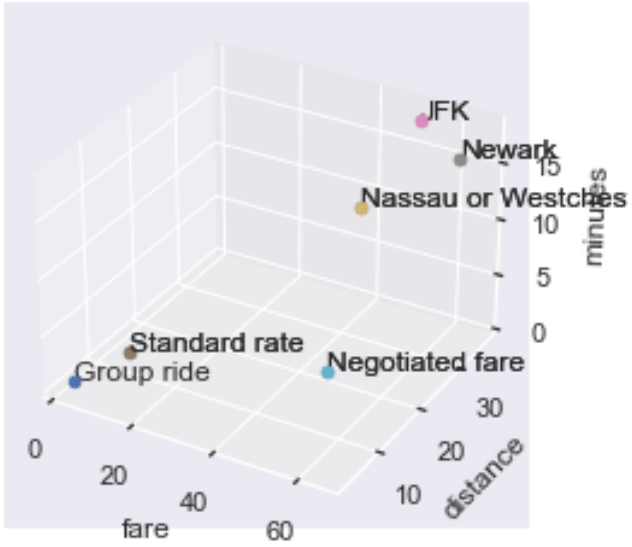

Рисунок 4.6 Визуализация медианных значений в пространстве

Заключение

В ходе работы были исследованы различные подходы анализа больших массивов данных, а также проведен их сравнительный анализ, выявлены преимущества и недостатки. Кроме того, были изучены различные методы оптимизации при обработке данных большого размера, например, такие, как стандартизация данных, удаление лишних признаков, преобразование типов и удаление выбросов.

Для лучшего понимания зависимостей между признаками были получены новые характеристики. Например, новые столбцы со значениями дня недели и точного часа совершения поездки были полезны при исследовании: в каких временных промежутках люди чаще всего вызывают такси и как от этого зависит стоимость поездки. Полученные результаты были визуализированы и проанализированы.

Таким образом, главная задача, заключающаяся в считывании данных, их предобработке, нахождении новых признаков и их визуализации, была выполнена. В результате проделанной работы данные были подготовлены для последующего машинного обучения с целью предсказания стоимости поездки на основе необходимых признаков.

Список использованной литературы

1. Pandas User Guide [Электронный ресурс] — Режим доступа: https://pandas.pydata.org/pandas-docs/stable/user_guide/index.html — Дата доступа: 01.04.2021
2. Vaex 4.1.0 Documentation [Электронный ресурс] — Режим доступа: https://vaex.readthedocs.io/en/latest/index.html — Дата доступа: 05.04.2021
3. Python и анализ данных / Уэс Маккинли. — М.: ДМК Пресс, 2015
4. Анализ данных: используем методы статистического исследования [Электронный ресурс] — Режим доступа: https://analytikaplus.ru/analiz-dannyh-statisticheskie-metody-issledovaniya/ — Дата доступа: 01.05.2021
5. Cleaning up data Data from Outliers [Электронный ресурс] — Режим доступа: https://www.pluralsight.com/guides/cleaning-up-data-from-outliers — Дата доступа: 01.05.2021
6. Основные понятия интеллектуального понятия [Электронный ресурс] — Режим доступа: https://docs.microsoft.com/ru-ru/analysis-services/data-mining/data-mining-concepts — Дата доступа: 07.05.2021
7. Wikipedia [Электронный ресурс] — Режим доступа: https://www.wikipedia.org/ — Дата доступа: 07.05.2021

Приложение