Курсовая работа:
тема: Обнаружение мошенничества с кредитными картами.

Содержание:



7. перспективы: улучшить безопасность залив все на AWS.
Улучшить гибкость работы используя airflow и spark
Сделать более глубокий анализ данных
Делать вычисления на GPU.

## Глава 0. Знакомство.

Добрый день, уважаемые преподаватели и товарищи по цеху. Я Пастушенко Антон и тема моей работы “Обнаружение мошенничества с кредитными картами”. На протяжении 7 минут я расскажу вам обо всем простым языком.

## Глава1. Введение.

Реальность.
В постоянно развивающемся современном мире люди пользуются банковскими картами. Все больше в нашу жизнь приходит безначичный расчет за услуги. Мы расплачиваемся картами в магазинах, кафе, заведениях, транспорте и так далее. Но большинство транзакций происходит через интернет. Абсолютное большинство денежных сделок происходят безналично. Просто перечисляют деньги на счет банка с другого счета. А реальные деньги просто лежат в хранилищах и выполняют другие функции. В данной работе будем дальше рассматривать только безналичные транзакции.

Наиболее распространенные сферы использования банковских карт/безналичного расчета:
1. Розничная торговля. Более 70% всех операций с картами относятся к этой сфере.
2. Онлайн-покупки. С каждым годом все больше людей делают покупки через интернет, поэтому онлайн-торговля становится все более популярной. Более 20% всех операций с банковскими картами относятся к онлайн-покупкам.

3. Путешествия: Оплата гостиниц, билетов на самолет и других туристических услуг - это еще одна распространенная сфера использования банковских карт. По статистике, около 7% всех операций с банковскими картами относятся к путешествиям.
4. Ресторанный бизнес: Оплата в ресторанах и кафе - это еще один вид операций, который становится все популярнее. По статистике, около 3% всех операций с банковскими картами относятся к ресторанным услугам.
5. Здравоохранение: Оплата медицинских услуг, страховок и лекарств также может производиться с помощью банковских карт. Этот вид операций составляет около 2% от всех операций с банковскими картами.
6. Развлечения: Билеты в кино, театр, концерты и другие развлекательные мероприятия также можно приобрести с помощью банковских карт. Они составляют около 1% всех операций с банковскими картами.

** пирог

## 1.1. Проблема и ее актуальность.

Так как эти операции проводятся через интернет, то к ним можно получить доступ и нанести тем самым ущерб. Сорвать сделку и многие люди потеряют работу. Не закупить оборудование и люди в больницах будут чувствовать себя плохо. Простое хищение средств, незаконный перевод на другой счет. Даже закупить, но не тот материал – и здание рухнет через год. Это были частные примеры. Чтобы ближе понять проблему. Если смотреть шире, то мошенничество может приводить к:
1. Финансовые потери: Когда карта крадется или ее данные компрометируются, мошенники могут использовать ее для совершения покупок и снятия денег со счета без разрешения владельца карты. Это может привести к серьезным финансовым потерям для клиента, банка и других организаций.

2. Ущерб репутации: Если банк не защищает своих клиентов от мошеннических операций, это может негативно повлиять на его репутацию. Клиенты могут потерять доверие к банку и перейти к конкурентам, что может привести к серьезным финансовым потерям.

3. Негативный влияние на экономику: Мошенничество с кредитными картами может привести к негативным последствиям для экономики в целом. Это может привести к увеличению стоимости кредитования и ухудшению кредитного рейтинга в стране.

4. Угроза безопасности: Кража или компрометация данных клиентов может привести к серьезной угрозе их безопасности. Кроме того, мошенничество с кредитными картами может быть связано с другими видами преступлений, такими как кража личных данных и кибератаки.

5. Увеличение расходов на безопасность: Для того, чтобы предотвратить мошенничество с кредитными картами, банки и другие организации вынуждены тратить большие деньги на усиление мер безопасности. Это может привести к увеличению расходов на

безопасность для банков и других организаций, что может отразиться на клиентах в виде увеличения комиссий и сборов.

На прямую влияет на качество жизни. Думаю, проблему мы уяснили.

## 1.2. Необходимость решения.

Чтобы не было перечисленнх выше проблем, нужно находить решения. Рассмотим уже существущие.

Уже сейчас активно находят решение проблем мошенничества. Банки внедряют новые технологии в свою систему транзакций.

1. Мониторинг транзакций: Банки могут мониторить транзакции, чтобы обнаружить любую необычную активность, такую как большие покупки, покупки из других стран или транзакции, произведенные в необычное время. Если банк обнаружит такую активность, он может связаться с владельцем карты, чтобы уточнить, была ли эта транзакция разрешена.

2. Системы защиты от мошенничества: Банки используют различные системы защиты от мошенничества, такие как 3D Secure, которые требуют от владельца карты ввести дополнительный пароль или код подтверждения при совершении онлайн-транзакции. Это помогает защитить карту от несанкционированных покупок.

3. Экспертная система анализа данных: Банки используют системы анализа данных, чтобы выявлять необычные паттерны транзакций и активности на карте. Эти системы могут быстро обнаруживать мошеннические схемы и предотвращать их до того, как они причинят серьезный ущерб.

4. Фильтрация данных: Банки также используют фильтрацию данных для защиты от мошенничества. Это может включать в себя фильтрацию IP-адресов или блокировку определенных типов транзакций, которые могут быть связаны с мошенничеством.

5. Использование технологии биометрии: Банки также могут использовать технологию биометрии, такую как сканирование отпечатков пальцев или распознавание лиц, чтобы подтвердить личность владельца карты. Это помогает предотвратить мошенничество, связанное с утерей карты или утечкой информации.

6. Внедрение новых технологий: Банки постоянно ищут новые технологии, которые могут помочь улучшить безопасность и защиту от мошенничества. Например, некоторые банки начали использовать блокчейн-технологии для обеспечения безопасности транзакций.

7. Сотрудничество с правоохранительными органами: Банки могут сотрудничать с правоохранительными органами, чтобы обнаружить и пресечь мошеннические операции, связанные с кредитными картами. Они могут обмениваться информацией и работать вместе для того, чтобы выявить мошенников и привлечь их к ответственности.

## 1.3. Мои цели и задачи.

Целью моей курсовой работы является разработка системы классификации транзакций в реальном времени на мошенническую или нет. Чтобы пользователь при совершении транзакции проходил определенную процедуру, которая оценивает надежность его транзакции. Затем можно передать эту оценку владельцу и банку для их дальнейших действий.

Для этого мне нужно работать с личными данными клиента и данными о его транзакции. Следовательно мне нужно обеспечить безопасность хранения и обработки данных.

Мне нужно выбрать значащие признаки из операций.

На основе их обучить модели бинарной классификации.

Мне нужно интерпретировать результаты, выбрать подход.

Сделать выводы о транзакции.

# 2. Инструменты для решения задачи.

Python – высокоуровневый язык программирования. Он является интерпретруемым языком со строгой динамической типизацией. Он широко используется в различных областях, таких как веб-разработка, научные вычисления, искусственный интеллект, машинное обучение, анализ данных, создание игр и многое другое. Так же он известен своими богатыми инструментариями для работы с данными, математикой и алгоритмами. Предоставляет удобный и гибкий интерфейс. Очень юогатая поддержка сообщества, постоянно развивается.

Numpy (Numerical Python) - это библиотека языка программирования Python для работы с числовыми массивами и матрицами. Она предоставляет мощные функции для работы с многомерными массивами, включая математические, логические, манипуляционные и сортировочные операции.

Она также предоставляет многие функции для выполнения математических операций, таких как линейная алгебра, трансформация Фурье и случайные числа.

Написана на C и Fortran – что существенно ускоряет ее работу

Pandas - библиотека языка программирования Python для обработки и анализа данных. Она предоставляет удобный и эффективный способ работы с таблицами данных предоставляет множество функций для работы с данными, включая сортировку, фильтрацию, группировку, объединение, агрегацию и многое другое. Она также предоставляет возможность чтения и записи данных из различных источников, таких как CSV, Excel, SQL-базы данных.

Matplotlib - это библиотека языка программирования Python для визуализации данных. Она позволяет создавать графики, диаграммы, гистограммы, рисунки и многое другое с помощью нескольких простых команд.

Seaborn - это библиотека визуализации данных на основе Matplotlib, которая позволяет создавать красивые и информативные графики и диаграммы с минимальным количеством кода. Seaborn предоставляет набор функций для создания различных типов графиков, таких как линейные графики, гистограммы, круговые диаграммы, диаграммы рассеяния и многое другое. Она также предоставляет множество параметров для настройки внешнего вида графиков

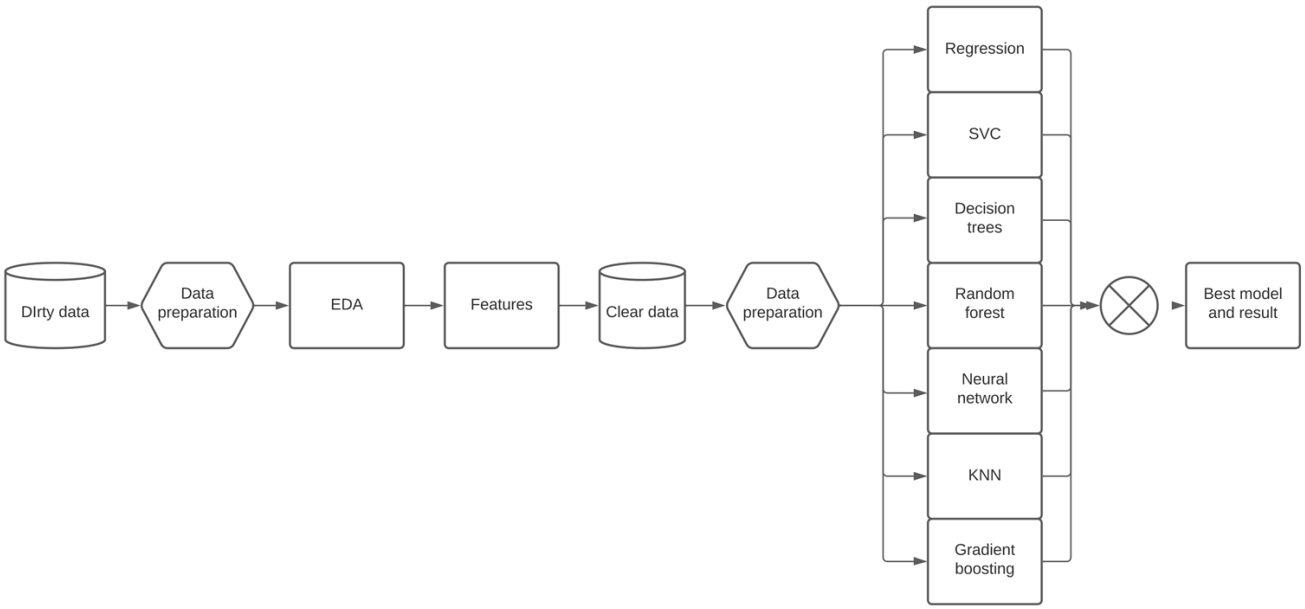
Scikit-learn - Scikit-learn (или sklearn) - это библиотека машинного обучения для языка программирования Python, которая предоставляет инструменты для решения задач классификации, регрессии, кластеризации и обработки данных. Scikit-learn базируется на библиотеках NumPy и SciPy, и является одной из наиболее популярных библиотек машинного обучения в Python.

Scikit-learn также предоставляет инструменты для предобработки данных, включая масштабирование, нормализацию, преобразование признаков и многое другое. Она также позволяет выполнить сбор и предварительную обработку данных из разных источников, включая файлы CSV, базы данных и т.д.

PostgreSQL - это мощная, открытая объектно-реляционная система управления базами данных. Она предоставляет широкие возможности для хранения и обработки структурированных данных, таких как таблицы, столбцы и строки. PostgreSQL поддерживает множество функций, включая транзакции, индексы, субдиаграммы, подзапросы, процедуры и триггеры. Она также обеспечивает безопасность данных с помощью авторизации пользователей, управления доступом и шифрования данных.

## 3. Архитектура приложения.

Рассмотрим архитектуру приложения, логический ход работы программы.




Имеется база данных с исходными данными о транзакциях пользователей.

Все остальные блоки представляют из себя python-скрипты, которые обрабатывают данные. Такие как:

1. Преобразование данных.
2. Расследовательный анализ данных.
3. Выделение признаков.
4. Запись признакового пространства в базу данных.
5. Подготовку данных к использованию моделями.
6. Моделирование и его оценка.
7. Решение поставленной задачи и его интерпретация.

## 3.1 База данных.

База данных:

В качестве хранилища конфиденциальных данных о транзакциях клиента используется PostgreSQL. Она имеет встроенные функции безопасности и возможности их настроить.

Кластер лежит локально на жестком диске. Используется одна база данных “NoNameBank”, таблица “train_transactions”, таблица “test_transactions” и пользователь “analytic” с правами:

- Создание таблиц.

- Чтение таблиц.

- Запись в им созданные таблицы.

Остальное ему запрещено. Подключение к базе данных с паролем, аутентификация пользователя с паролем, подключение к таблицам “transactions” с паролем.

Данные надежно защищены и соответствуют правилам обработки персональных данных.

## 3.2 EDA

Блок EDA основной по важности. Его рассмотрим далее во всех красках и графиках. В нем рассматриваются сами данные и ищутся зависимости между данными и целевой переменной – фактом мошенничества.

## 3.3 Модели.

В данной работе для решения задачи классификации использовались следующие алгоритмы машинного обучения:

1. Логистическая регрессия - это статистическая модель, которая используется для моделирования вероятности наступления определенного события. Логистическая регрессия применяется для бинарной классификации, где выходом является бинарное значение, например, 0 или 1.

2. Метод опорных векторов – это метод, который строит гиперплоскость или несколько гиперплоскостей в пространстве высокой размерности для разделения данных на классы. SVM также может быть использован для многоклассовой классификации.
Может работать с несбалансированными данными.

3. Решающие деревья - это модель, которая использует деревья для принятия решений на основе различных признаков. Решающие деревья также могут быть использованы для многоклассовой классификации.

4. Случайный лес – может работать с несбалансированными данными. Лучше сбалансировать. Случайный лес также может быть использован для задач бинарной классификации. Для этого, каждый дерево строится на случайном подмножестве данных и признаков, а затем все деревья комбинируются в один классификатор.
В задачах бинарной классификации, случайный лес может быть особенно полезен, так как он способен обрабатывать данные с большим количеством признаков, а также может работать с различными типами признаков. Кроме того, случайный лес может обработать несбалансированные данные, когда количество примеров в одном классе значительно превышает количество примеров в другом.

5. KNN - это алгоритм, который определяет класс объекта на основе его близости к другим объектам. Он используется для поиска K ближайших объектов и принятия решения на основе того, какие классы у этих объектов.

Основываясь на приведенном выше EDA, мы обнаружили, что такие функции, как сумма транзакции, возраст держателя кредитной карты, категория расходов, время транзакции и местоположение, имеют разную степень корреляции с мошенничеством с кредитными картами. Это помогает нам выбрать, какие функции мы хотим включить в наши модели данных. План состоит в том, чтобы обучить модели на наборе обучающих данных, который мы проанализировали выше, а затем использовать набор тестовых данных для оценки производительности модели.
- Плохо

6. Бустинг над решающими деревьями == Градиентный бустинг. -
это метод машинного обучения, который использует ансамбль из решающих деревьев для улучшения точности прогнозирования. Он основан на идее последовательного построения слабых моделей, каждая из которых нацелена на исправление ошибок предыдущих моделей.

Заключительным этапом работы является ответ на поставленную задачу, решение поставленной проблемы, оценка этого решения и его интерпретация.

## 4. Доступ к данным.

Подключаемя к таблице “transactions” пользователем “analytic”.

Признаки, которые описывают транзакцию:

1. index – уникальный номер транзакции. int

2. trans_date_trans_time – дата и время проведения транзакции. object

3. cc_num – номер кредитной карты клиента.

4. merchant – наименование продавца.

5. category – категория товара.

6. amt – сумма транзакции.

7. first – имя держателя карты.

8. last – фамилия держателя карты.

9. gender – пол держателя карты.

10. street – адрес держателя карты.

11. city – город держателя карты.

12. state – штат проживания держателя карты.

13. zip – почтовый индекс держателя карты.

14. lat – географическая широта держателя карты.

15. long - географическая долгота держателя карты.

16. city_pop – население города проживания.

17. job – рабочая должность держателя карты.

18. dob – дата рождения держателя карты.

19. trans_num – номер транзакции.

20. unix_time – время совершения транзакции в формате unix.

21. merch_lat – географическая широта продавца.

22. merch_long - географическая долгота продавца.

23. is_fraud – факт мошенничества.

## 5. Очистка данных.

Произведем загрузку таблицы в датафрейм пандас. И проведем очистку:

1. Удалим дубликаты.

2. Посмотрим на количество нулевых значений в таблице:

Их 0

3. Посмотрим на описание датафрейма и узнаем типы данных его колонок:

```
Data columns (total 23 columns):
 #   Column                 Non-Null Count    Dtype  
---  ------                 --------------    -----  
 0   Unnamed: 0             1296675 non-null  int64  
 1   trans_date_trans_time  1296675 non-null  object 
 2   cc_num                 1296675 non-null  int64  
 3   merchant               1296675 non-null  object 
 4   category               1296675 non-null  object 
 5   amt                    1296675 non-null  float64
 6   first                  1296675 non-null  object 
 7   last                   1296675 non-null  object 
 8   gender                 1296675 non-null  object 
 9   street                 1296675 non-null  object 
 10  city                   1296675 non-null  object 
 11  state                  1296675 non-null  object 
 12  zip                    1296675 non-null  int64  
 13  lat                    1296675 non-null  float64
 14  long                   1296675 non-null  float64
 15  city_pop               1296675 non-null  int64  
 16  job                    1296675 non-null  object 
 17  dob                    1296675 non-null  object 
 18  trans_num              1296675 non-null  object 
 19  unix_time              1296675 non-null  int64  
 20  merch_lat              1296675 non-null  float64
 21  merch_long             1296675 non-null  float64
 22  is_fraud               1296675 non-null  int64  
dtypes: float64(5), int64(6), object(12)
memory usage: 237.4+ MB
```

4. Переименуем колонки:

trans_date_trans_time' – 'transaction_time'
'cc_num' – 'card_number'
'amt' – 'amount(usd)',
'trans_num' – 'transaction_id'
удалим колонку индексов.
Создадим новые колонки:

time – unix дата и время
hour_of_day – час суток.
Age – возраст держателя карты.
day_of_week – день недели.

То же самое проделали и с тестовой выборкой.

## 6. EDA.