Geekbrains

**Прогнозирование аналитических данных бухгалтерского учета по текстовому описанию платежного документа**

Искусственный интеллект

Машинное обучение

Проторский А.В.

Москва

2024

**Содержание**

# Введение

**Тема проекта:** Обучить модель машинного обучения для предсказания аналитических параметров бухгалтерского учета по текстовому описанию платежного документа.

**Цель:** Автоматизировать формирования бухгалтерских документов при обработке банковских выписок, сократив труд программистов и сотрудников бухгалтерии при загрузке данных из системы клиент-банк в бухгалтерскую систему.

**Какую проблему решает:** При формировании бухгалтерских документов из платежных необходимо определять параметры аналитического учета, для чего требуется проанализировать текстовое назначение платежа, для этого используется разбор текста с помощью регулярных выражений и человеческий труд по корректировки ошибочно определенных данных с последующим изменением алгоритма их разбора. Поскольку текст платежного поручения заполняется сотрудником банка в произвольной форме процесс модификации алгоритма постоянно продолжается, чтобы минимизировать время сопровождения системы, было принято решения обучить модель для решения данной задачи.

**Задачи:**

1. Изучить литературу, касающуюся темы исследования.
2. Рассмотреть основные виды и методы моделей машинного обучения.
3. Разработать и обучить модель машинного обучения.
4. Создать WEB REST API сервис для использования обученной модели.

**Инструменты:** Python**,** Jupyter Notebook,VSCode, Sklearn, FLASK, SoupUI, Git, WSL, Docker, Gunicorn

**Состав команды**: Проторский А.В. (Разработчик)

# Теоретическая часть

# Введение

Благодаря машинному обучению программист не обязан писать инструкции, учитывающие все возможные проблемы и содержащие все решения. Вместо этого в программу закладывают алгоритм самостоятельного нахождения решений путём комплексного использования статистических данных, из которых выводятся закономерности и на основе которых делаются прогнозы.

Технология машинного обучения на основе анализа данных берёт начало в 1950 году, когда начали разрабатывать первые программы для игры в шашки. За прошедшие десятилетий общий принцип не изменился. Зато благодаря взрывному росту вычислительных мощностей компьютеров многократно усложнились закономерности и прогнозы, создаваемые ими, и расширился круг проблем и задач, решаемых с использованием машинного обучения.

Чтобы запустить процесс машинного обучение, для начала необходимо загрузить в компьютер набор данных, на которых алгоритм будет учиться обрабатывать запросы. Например, могут быть фотографии собак и котов, на которых уже есть метки, обозначающие к кому они относятся. После процесса обучения, программа уже сама сможет распознавать собак и котов на новых изображениях без содержания меток. Процесс обучения продолжается и после выданных прогнозов, чем больше данных мы проанализировали программой, тем более точно она распознает нужные изображения.

Благодаря машинному обучению компьютеры учатся распознавать на фотографиях и рисунках не только лица, но и пейзажи, предметы, текст и цифры. Что касается текста, то и здесь не обойтись без машинного обучения: функция проверки грамматики сейчас присутствует в любом текстовом редакторе и даже в телефонах. Причем учитывается не только написание слов, но и контекст, оттенки смысла и другие тонкие лингвистические аспекты.

# Типы задач машинного обучения

Все задачи, решаемые с помощью машинного обучения, относятся к одной из следующих категорий.

1. Задача регрессии – прогноз на основе выборки объектов с различными признаками. На выходе должно получиться вещественное число (2, 35, 76.454 и др.), к примеру цена квартиры, стоимость ценной бумаги по прошествии полугода, ожидаемый доход магазина на следующий месяц, качество вина при слепом тестировании.
2. Задача классификации – получение категориального ответа на основе набора признаков. Имеет конечное количество ответов (как правило, в формате «да» или «нет»): есть ли на фотографии кот, является ли изображение человеческим лицом..
3. Задача кластеризации – распределение данных на группы: разделение всех клиентов мобильного оператора по уровню платёжеспособности, отнесение космических объектов к той или иной категории (планета, звёзда, чёрная дыра и т. п.).
4. Задача уменьшения размерности – сведение большого числа признаков к меньшему (обычно 2–3) для удобства их последующей визуализации (например, сжатие данных).
5. Задача выявления аномалий – отделение аномалий от стандартных случаев. На первый взгляд она совпадает с задачей классификации, но есть одно существенное отличие: аномалии – явление редкое, и обучающих примеров, на которых можно обучить модель на выявление таких объектов, либо исчезающе мало, либо просто нет, поэтому методы классификации здесь не работают. На практике такой задачей является, например, выявление мошеннических действий с банковскими картами.

# Основные виды машинного обучения

Основная масса задач, решаемых при помощи методов машинного обучения, относится к двум разным видам: обучение с учителем (supervised learning) либо без него (unsupervised learning). Однако этим учителем вовсе не обязательно является сам программист, который стоит над компьютером и контролирует каждое действие в программе. «Учитель» в терминах машинного обучения – это само вмешательство человека в процесс обработки информации. В обоих видах обучения машине предоставляются исходные данные, которые ей предстоит проанализировать и найти закономерности. Различие лишь в том, что при обучении с учителем есть ряд гипотез, которые необходимо опровергнуть или подтвердить.

1. **Машинное обучение с учителем**

Предположим, в нашем распоряжении оказались сведения о десяти тысячах московских квартир: площадь, этаж, район, наличие или отсутствие парковки у дома, расстояние от метро, цена квартиры и т. п. Нам необходимо создать модель, предсказывающую рыночную стоимость квартиры по её параметрам. Это идеальный пример машинного обучения с учителем: у нас есть исходные данные (количество квартир и их свойства, которые называются признаками) и готовый ответ по каждой из квартир – её стоимость(целевая переменная). Программе предстоит решить задачу регрессии, предсказать цену квартиры в зависимости от ее признаков.

1. **Машинное обучение без учителя**

В случае обучения без учителя, когда готовых «правильных ответов» системе не предоставлено, всё обстоит ещё интереснее. Например, у нас есть информация о весе и росте какого-то количества людей, и эти данные нужно распределить по трём группам, для каждой из которых предстоит пошить рубашки подходящих размеров. Это задача кластеризации. В этом случае предстоит разделить все данные на 3 кластера (но, как правило, такого строгого и единственно возможного деления нет). Если взять другую ситуацию, когда каждый из объектов в выборке обладает сотней различных признаков, то основной трудностью будет графическое отображение такой выборки. Поэтому количество признаков уменьшают до двух или трёх, и становится возможным визуализировать их на плоскости или в 3D.

Это – задача уменьшения размерности.

# Основные алгоритмы моделей машинного обучения

Машинное обучение использует широкий спектр алгоритмов для перевода наборов данных в прогнозные модели. Какой алгоритм сработает лучше, зависит от решаемой задачи.

* + - 1. **Дерево принятия решений**

Это метод поддержки принятия решений, основанный на использовании древовидного графа: модели принятия решений, которая учитывает их потенциальные последствия (с расчётом вероятности наступления того или иного события), эффективность, ресурсозатратность. Для бизнес-процессов это дерево складывается из минимального числа вопросов, предполагающих однозначный ответ — «да» или «нет». Последовательно дав ответы на все эти вопросы, мы приходим к правильному выбору. Методологические преимущества дерева принятия решений – в том, что оно структурирует и систематизирует проблему, а итоговое решение принимается на основе логических выводов.

* + - 1. **Наивная байесовская классификация**

Наивные байесовские классификаторы относятся к семейству простых вероятностных классификаторов и берут начало из теоремы Байеса, которая применительно к данному случаю рассматривает функции как независимые (это называется строгим, или наивным, предположением).

* + - 1. **Метод наименьших квадратов (линейная регрессия)**

Всем, кто хоть немного изучал статистику, знакомо понятие линейной регрессии. К вариантам её реализации относятся и наименьшие квадраты. Обычно с помощью линейной регрессии решают задачи по подгонке прямой, которая проходит через множество точек. Вот как это делается с помощью метода наименьших квадратов: провести прямую, измерить расстояние от неё до каждой из точек (точки и линию соединяют вертикальными отрезками), получившуюся сумму перенести наверх. В результате та кривая, в которой сумма расстояний будет наименьшей, и есть искомая (эта линия пройдёт через точки с нормально распределённым отклонением от истинного значения). Линейная функция обычно используется при подборе данных для машинного обучения, а метод наименьших квадратов – для сведения к минимуму погрешностей путем создания метрики ошибок.



Задача модели - в процессе обучения подобрать коэффициенты ω так, чтобы значение этой функции было как можно более близко к реальному целевому значению y объекта x.

* + - 1. **Логистическая регрессия**

Логистическая регрессия – это способ определения зависимости между переменными, одна из которых категориально зависима, а другие независимы. Для этого применяется логистическая функция (аккумулятивное логистическое распределение). Практическое значение логистической регрессии заключается в том, что она является мощным статистическим методом предсказания событий, который включает в себя одну или несколько независимых переменных, по которым предсказывается вероятность принадлежности данного объекта к каждому из классов.

Рассмотрим алгоритм логистической регрессии для бинарной классификации, т.е. случая, когда различных классов всего 2. В этом случае задача алгоритма логистической регрессии очень похожа на задачу алгоритма линейной регрессии, рассмотренного ранее: модель должна подобрать коэффициенты ω0, ω1,… ωn. , затем определяется значение:



, величина z помещается в сигмоидную функцию:



Сигмоидная функция используется для того, чтобы преобразовать число z из промежутка (−∞,∞) в число f(z) из промежутка (0,1). Причём, делается это так, что число f(z) можно теперь интерпретировать как вероятность. В нашем случае это будет вероятность принадлежности объекта x классу 1. Соответственно, вероятность принадлежности объекта классу 0 будет равна 1− f(z).

Алгоритм логистической регрессии минимизирует функцию потерь:



Здесь y - правильный ответ на объекте x (0 или 1), а p - вычисленная алгоритмом вероятность принадлежности объекта x классу 1.

* + - 1. **Метод опорных векторов (SVM)**

Это целый набор алгоритмов, необходимых для решения задач на классификацию и регрессионный анализ. Исходя из того что объект, находящийся в N-мерном пространстве, относится к одному из двух классов, метод опорных векторов строит гиперплоскость с мерностью (N – 1), чтобы все объекты оказались в одной из двух групп. На бумаге это можно изобразить так: есть точки двух разных видов, и их можно линейно разделить. Кроме сепарации точек, данный метод генерирует гиперплоскость таким образом, чтобы она была максимально удалена от самой близкой точки каждой группы.

* + - 1. **Метод ансамблей**

Он базируется на алгоритмах машинного обучения, генерирующих множество классификаторов и разделяющих все объекты из вновь поступающих данных на основе их усреднения или итогов голосования. Изначально метод ансамблей был частным случаем байесовского усреднения, но затем усложнился и оброс дополнительными алгоритмами:

* бустинг (boosting) – преобразует слабые модели в сильные посредством формирования ансамбля классификаторов (с математической точки зрения это является улучшающим пересечением);
* бэггинг (bagging) – собирает усложнённые классификаторы, при этом параллельно обучая базовые (улучшающее объединение);
* корректирование ошибок выходного кодирования.

Метод ансамблей – более мощный инструмент по сравнению с отдельно стоящими моделями прогнозирования, поскольку:

* он сводит к минимуму влияние случайностей, усредняя ошибки каждого базового классификатора;
* уменьшает дисперсию, поскольку несколько разных моделей, исходящих из разных гипотез, имеют больше шансов прийти к правильному результату, чем одна отдельно взятая;
* исключает выход за рамки множества: если агрегированная гипотеза оказывается вне множества базовых гипотез, то на этапе формирования комбинированной гипотезы оно расширяется при помощи того или иного способа, и гипотеза уже входит в него.
  + - 1. **Алгоритмы кластеризации**

Кластеризация заключается в распределении множества объектов по категориям так, чтобы в каждой категории – кластере – оказались наиболее схожие между собой элементы.  
  
Кластеризировать объекты можно по разным алгоритмам. Чаще всего используют следующие:

* на основе центра тяжести треугольника;
* на базе подключения;
* сокращения размерности;
* плотности (основанные на пространственной кластеризации);
* вероятностные;
* машинное обучение, в том числе нейронные сети.
  + - 1. **Метод главных компонент (PCA)**

Метод главных компонент, или PCA, представляет собой статистическую операцию по ортогональному преобразованию, которая имеет своей целью перевод наблюдений за переменными, которые могут быть как-то взаимосвязаны между собой, в набор главных компонент – значений, которые линейно не коррелированы. Практические задачи, в которых применяется PCA, – визуализация и большинство процедур сжатия, упрощения, минимизации данных для того, чтобы облегчить процесс обучения. Однако метод главных компонент не годится для ситуаций, когда исходные данные слабо упорядочены (то есть все компоненты метода характеризуются высокой дисперсией). Так что его применимость определяется тем, насколько хорошо изучена и описана предметная область.

* + - 1. **Стохастическое вложение соседей с t-распределением**

Алгоритм t-SNE (t-distributed Stochastic Neighbor Embedding) позволяет понижать размерность данных до двух или трёх измерений, что позволяет визуализировать данные на двумерных и трёхмерных графиках. Изучая графики, можно, например, понять, на сколько кластеров адекватно разбивать данные, а также оценить уже выполненное разбиение на кластеры.

* + - 1. **Сингулярное разложение**

В линейной алгебре сингулярное разложение, или SVD, определяется как разложение прямоугольной матрицы, состоящей из комплексных или вещественных чисел. Так, матрицу M размерностью [m\*n] можно разложить таким образом, что M = UΣV, где U и V будут унитарными матрицами, а Σ – диагональной. Одним из частных случаев сингулярного разложения является метод главных компонент. Самые первые технологии компьютерного зрения разрабатывались на основе SVD и PCA и работали следующим образом: вначале лица (или другие паттерны, которые предстояло найти) представляли в виде суммы базисных компонент, затем уменьшали их размерность, после чего производили их сопоставление с изображениями из выборки. Современные алгоритмы сингулярного разложения в машинном обучении, конечно, значительно сложнее, чем их предшественники, но суть их в целом нем изменилась.

* + - 1. **Анализ независимых компонент (ICA)**

Это один из статистических методов, который выявляет скрытые факторы, оказывающие влияние на случайные величины, сигналы и пр. ICA формирует порождающую модель для баз многофакторных данных. Переменные в модели содержат некоторые скрытые переменные, причем нет никакой информации о правилах их смешивания. Эти скрытые переменные являются независимыми компонентами выборки и считаются негауссовскими сигналами.  
  
В отличие от анализа главных компонент, который связан с данным методом, анализ независимых компонент более эффективен, особенно в тех случаях, когда классические подходы оказываются бессильны. Он обнаруживает скрытые причины явлений и благодаря этому нашёл широкое применение в самых различных областях – от астрономии и медицины до распознавания речи, автоматического тестирования и анализа динамики финансовых показателей

# Основные метрики оценки качества моделей машинного обучения

Метрики оценки качества моделей разделяются для задач классификации и регрессии.

* + - 1. **Оценка качества модели в задачах классификации**

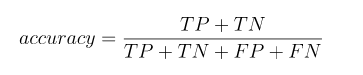
Перед переходом к самим метрикам необходимо ввести важную концепцию для описания этих метрик в терминах ошибок классификации — confusion matrix (матрица ошибок).

|  |  |  |
| --- | --- | --- |
|  | y = 1 | y = 0 |
| Ὗ = 1 | True Positive (TP) | False Positive (FP) |
| Ὗ = 0 | False Negative (FN) | True Negative (TN) |

Здесь Ὗ - это ответ алгоритма на объекте, а y - истинная метка класса на этом объекте.  
Таким образом, ошибки классификации бывают двух видов: False Negative (FN) и False Positive (FP).

#### **Accuracy**

Доля правильных ответов модели рассчитывается по формуле:



Использование данной метрики не дает достоверную оценку качества модели, т.к при прогнозировании всех событий как TN будет получена более высокая оценка чем в иных случаях, следовательно модель не обладает ни какой предсказательной силой.

#### **Precision, recall и F-мера**

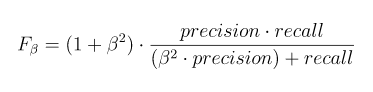
Для оценки качества работы алгоритма на каждом из классов по отдельности введем метрики precision (точность) и recall (полнота).

|  |  |
| --- | --- |
|  |  |

Precision можно интерпретировать как долю объектов, названных классификатором положительными и при этом действительно являющимися положительными, а recall показывает, какую долю объектов положительного класса из всех объектов положительного класса нашел алгоритм.

Именно введение precision не позволяет нам записывать все объекты в один класс, так как в этом случае мы получаем рост уровня False Positive. Recall демонстрирует способность алгоритма обнаруживать данный класс вообще, а precision — способность отличать этот класс от других классов. Precision и recall не зависят, в отличие от accuracy, от соотношения классов и потому применимы в условиях несбалансированных выборок.

Существует несколько различных способов объединить precision и recall в агрегированный критерий качества. F-мера, среднее гармоническое precision и recall :



Β в данном случае определяет вес точности в метрике, при Β = 1 это среднее гармоническое (с множителем 2, чтобы в случае precision = 1 и recall = 1 иметь F = 1). F-мера достигает максимума при полноте и точности, равными единице, и близка к нулю, если один из аргументов близок к нулю.

Если важность метрик Precision и Recall одинакова, можно использовать F1 – мерой:



Необходимо отметить, что в случае задач с несбалансированными классами, которые превалируют в реальной практике, часто приходится прибегать к техникам искусственной модификации набора данных для выравнивания соотношения классов, такие как oversampling (увеличение количество наблюдений с меньшим классом) и undersampling (уменьшение количества наблюдений доминирующего класса)

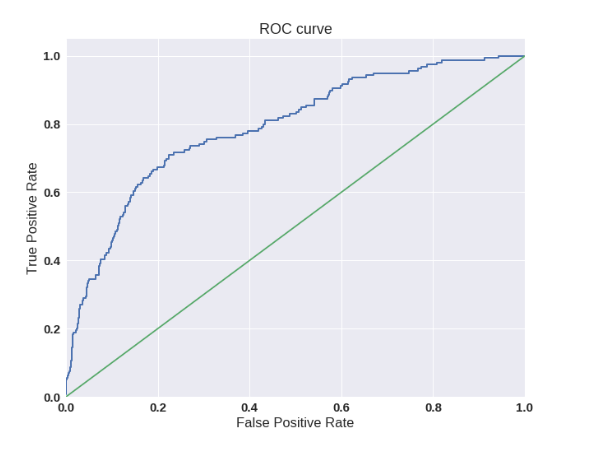
#### **AUC-ROC**

При конвертации вещественного ответа алгоритма (как правило, вероятности принадлежности к классу) в бинарную метку, мы должны выбрать какой-либо порог, при котором 0 становится 1. Естественным и близким кажется порог, равный 0.5, но он не всегда оказывается оптимальным, например, при вышеупомянутом отсутствии баланса классов.

Одним из способов оценить модель в целом, не привязываясь к конкретному порогу, является AUC-ROC (или ROC AUC) — площадь (*A*rea *U*nder *C*urve) под кривой ошибок (*R*eceiver *O*perating *C*haracteristic curve ). Данная кривая представляет из себя линию от (0,0) до (1,1) в координатах True Positive Rate (TPR) и False Positive Rate (FPR):

|  |  |
| --- | --- |
|  |  |

TPR это полнота, а FPR показывает, какую долю из объектов negative класса алгоритм предсказал неверно. В идеальном случае, когда классификатор не делает ошибок (FPR = 0, TPR = 1) мы получим площадь под кривой, равную единице; в противном случае, когда классификатор случайно выдает вероятности классов, AUC-ROC будет стремиться к 0.5, так как классификатор будет выдавать одинаковое количество TP и FP.



Недостатком данного метода является то, что AUC-ROC измеряет долю False Positive относительно True Negative и в задачах, где нам не так важен второй (больший) класс, может давать не совсем адекватную картину при сравнении алгоритмов.

Установка

wsl –install

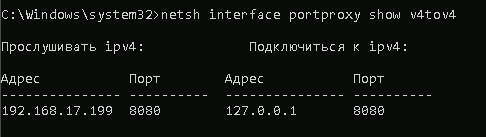
wsl –version

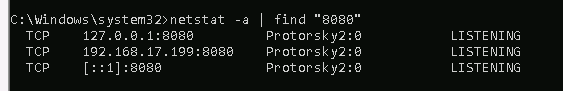
литература

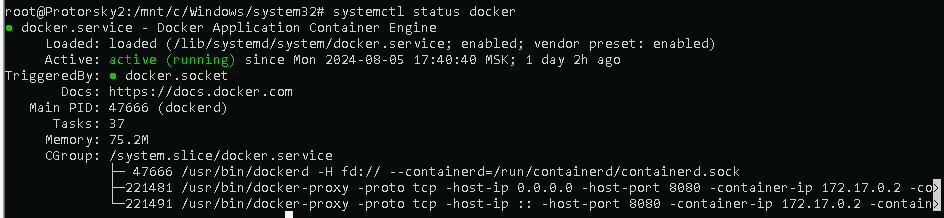
<https://habr.com/ru/companies/vdsina/articles/496804/>

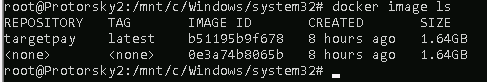
|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

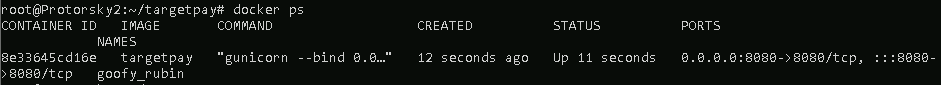
netsh interface portproxy add v4tov4 listenport=8080 listenaddress=192.168.17.199 connectport=8080 connectaddress=127.0.0.1











# Обновляем список пакетов

$ sudo apt-get update

# Устанавливаем необходимые пакеты, которые позволяют apt получить доступ по HTTPS

$ sudo apt-get install apt-transport-https ca-certificates curl gnupg2 software-properties-common

# Добавляем в свою систему ключ GPG официального репозитория Docker

$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

#Добавляем репозиторий Docker в список пакетов:

$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable"

# Обновляем данные о пакетах с учетом пакетов Docker из вновь добавленного репозитория

$ sudo apt-get update

# Установка Docker

$ sudo apt-get install docker-ce=17.09.0~ce-0~ubuntu

#запуска сервиса Docker

$ sudo nano /usr/local/sbin/start\_docker.sh

#!/usr/bin/env bash

sudo cgroups-mount

sudo service docker start

#Теперь разрешите скрипту запуститься и выполните его:

$ sudo chmod +x /usr/local/sbin/start\_docker.sh

# Запрещаем запись

$ sudo chmod 755 /usr/local/sbin/start\_docker.sh

$ /bin/sh /usr/local/sbin/start\_docker.sh

# Сборка

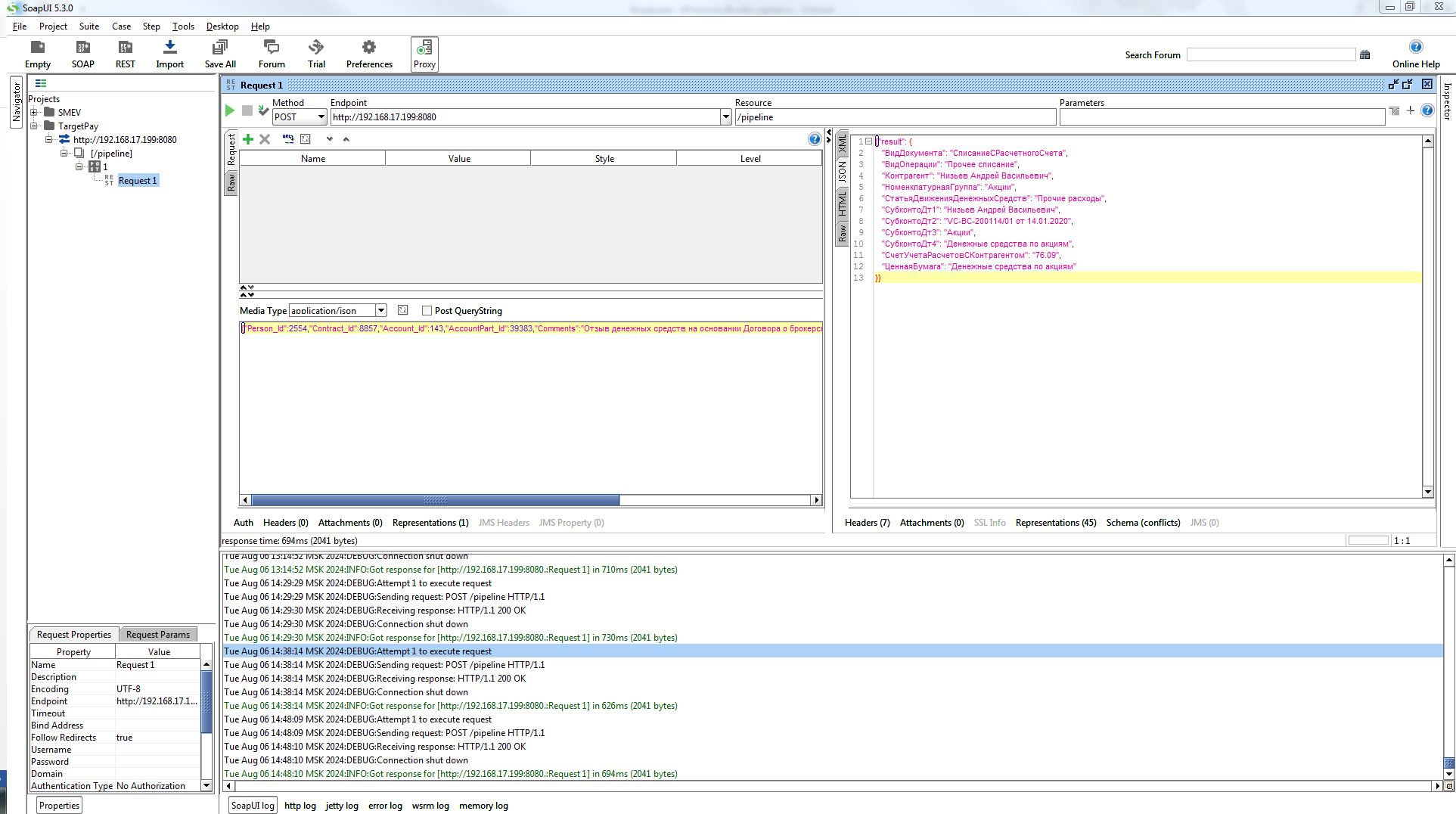
docker build -t targetpay .

# Запуск

docker run -d -p 8080:8080 targetpay

# Доступ из вне

netsh interface portproxy add v4tov4 listenport=8080 listenaddress=192.168.17.199 connectport=8080 connectaddress=127.0.0.1



{"Person\_Id":2554,"Contract\_Id":8857,"Account\_Id":143,"AccountPart\_Id":39383,"Comments":"Отзыв денежных средств на основании Договора о брокерском обслуживании № VC-BC-200114/ 01 от 14.01.2020. НДС не облагается."}

{"result": {

"ВидДокумента": "СписаниеСРасчетногоСчета",

"ВидОперации": "Прочее списание",

"Контрагент": "Низьев Андрей Васильевич",

"НоменклатурнаяГруппа": "Акции",

"СтатьяДвиженияДенежныхСредств": "Прочие расходы",

"СубконтоДт1": "Низьев Андрей Васильевич",

"СубконтоДт2": "VC-BC-200114/01 от 14.01.2020",

"СубконтоДт3": "Акции",

"СубконтоДт4": "Денежные средства по акциям",

"СчетУчетаРасчетовСКонтрагентом": "76.09",

"ЦеннаяБумага": "Денежные средства по акциям"

}}

Литература

<https://learn.microsoft.com/ru-ru/windows/wsl/tutorials/wsl-containers>

<https://www.dmosk.ru/miniinstruktions.php?mini=docker-self-image#newimage>

<https://habr.com/ru/articles/548910/>

<https://scikit-learn.org/stable/>