

**Федеральное государственное автономное образовательное учреждение
высшего образования «Национальный исследовательский университет
«Высшая школа экономики»**

Факультет компьютерных наук

Магистерская программа Финансовые технологии и анализ данных

Департамент _____

КУРСОВАЯ РАБОТА

**На тему «Разработка инструмента для мониторинга и анализа состояния
портфеля кредитных требований банка»**

Студент группы № мФТиАД
Борисенко Глеб Витальевич

Руководитель КР
Академический руководитель
Масютин Алексей Александрович

Консультант
Руководитель проектов
Щербаков Игорь Андреевич

Москва, 2020

Для содержания

Для реферата

Введение

Одним из этапов исследования данных на сегодняшний день является визуальный анализ. Он проводится с помощью различного рода диаграмм и графиков. Такой анализ позволяет выявлять закономерности, находить аномалии, и многое другое. Также после введения модели в эксплуатацию одним из важнейших для поддержания ее работы процессов является мониторинг показателей.

В качестве инструмента для визуального анализа и мониторинга показателей часто используется язык программирования с соответствующими библиотеками, например, язык Python с пакетом `seaborn`. Очевидным преимуществом такого подхода является высокая гибкость, но расплачиваться за это приходится достаточно большим количеством времени, необходимого для создания информативного графика и внесения даже минорных изменений.

Помимо описанного выше подхода, можно использовать специализированное ПО. Оно позволяет снизить порог вхождения благодаря отсутствию необходимости в знании программного API, упростить создание и изменение диаграмм с помощью графического интерфейса, а также решает большое количество неявных проблем. Но в все эти преимущества идут рука об руку с весомыми ограничениями. Например, поддержка только определенного формата, всего несколько возможных типов графиков, дополнительные ограничения, вызванные интерфейсом.

В моей работе предлагается совместить оба подхода для создания программного решения, обладающего изложенными выше плюсами и лишенного если не всех, то хотя бы нескольких недостатков, с целью создать интерактивную панель с диаграммами, позволяющую обеспечить простой и удобный мониторинг определенных показателей кредитного портфеля.

1 Постановка задачи

Целью работы является с помощью разработанного программного решения сконструировать интерактивный дашборд – панель с диаграммами. Ниже представлены требования как для ПО, так и для дашборда.

В качестве источника данных изначально должен поддерживаться формат csv как наиболее распространенный, но дальнейшее добавление новых источников должно быть простым и понятным. Должна существовать возможность разместить панель диаграмм на сервере, к которому можно подключаться с удаленного клиента.

Все диаграммы дашборда показателей должны поддерживать фильтрацию по дате. Диаграммы должны отображать следующие показатели:

- Численность договоров – количество договоров на дату с учётом всех фильтров в виде линейного графика
- Численность клиентов – количество клиентов на дату с учётом всех фильтров в виде линейного графика
- Default rate – доля тех, кто выйдет в дефолт в течение года, от тех, кто не в дефолте на дату оценки в виде линейного графика
- Уровень модельных оценок PD – среднее значение по полю вероятности дефолта с учётом всех фильтров. Исторический период – в виде линейного графика, а также барометр для самой последней даты.
- Распределение по рейтингам - столбчатая диаграмма
- Качество модели - показатель Gini.

2 Разработка платформы конструирования дашборда

2.1 Поиск и сравнение существующих решений

Одним из ключевых атрибутов системы должна быть открытость, т.е. это должно быть open source решение. Это позволит добиться той же гибкости, что дает использование только языка программирования. Из-за этого сразу отпадают лучшие приложения для исследования и визуализации данных: Qlik и Tableau.

Среди open source платформ наиболее популярная и удобная это Grafana. Она позволяет создавать интерактивные дашборды с большим набором возможных типов диаграмм и источников данных, а плагины сообщества дополнительно расширяют функционал. Главным недостатком с точки зрения данной работы является отсутствие поддержки в качестве источника данных внешнего файл – все предлагаемые источники — это различные базы данных. Но Grafana обладает достаточно понятным API, что позволяет самому создать плагин. Именно это стало основополагающим фактором в выборе платформы конструирования дашборда.

2.2 Обзор Grafana

Grafana позволяет создавать интерактивные дашборды. Пример такого дашборда[1] представлен на рисунке 1. Каждый дашборда состоит из набора панелей, каждая из которых и является диаграммой. Есть несколько видов диаграмм, которые платформа поддерживает прямо из коробки, но огромное количество типов доступны в качестве плагинов, которые просто ставятся через интерфейс командной строки либо обычным перемещением папок. Также среди визуализаций есть различные варианты текстового представления обычных или критических данных, вроде таблицы, списка тревог или даже графической схемы.

Ниже представлен список типов графиков, которые поддерживаются Grafana прямо из коробки:

- Линейный график
- Гистограмма
- Столбчатая диаграмма
- Барометр
- Тепловая карта
- Круговая диаграмма



Рисунок 1- Пример дашборда Grafana

У каждой панели, в зависимости от выбранного типа визуализации, есть множество настроек, позволяющее создавать диаграммы на свой вкус и цвет. Например, для линейного графика можно регулировать формат линий, осей, легенду, добавлять границы для цветного отображения, временные регионы,

ссылки и другое. Каждая панель интерактивная, ее можно переместить, трансформировать и поделиться.

Grafana работает в основном с временными рядами, т.е. датафреймами с индексом по дате, поэтому в ней сразу есть фильтрация по дате. Есть возможность работы и с табличными данными без дат, но в таком случае визуализация доступна только для агрегаций над массивами. К счастью, эта проблема решается сторонними плагинами, например, плагином Plotly[2].

Для диаграмм нужны данные, по которым их надо рисовать, а данные Grafana получает путем отправки запроса к источнику данных. Список поддерживаемых источников достаточно обширен, каждый из них представлен в виде плагина. У каждого ресурса свой формат запроса, и в большинстве это запрос на языке SQL, т.к. это самое большинство – вариации баз данных. Из обнаруженных минусов – на одной панели не может быть графиков, полученных с помощью разных источников, но при этом возможно создавать несколько запросов, таким образом отображая сразу несколько массивов. Помимо этого, платформа позволяет настроить обновление панелей путем посылки повторного запроса с заданной частотой.

Также Grafana позволяет добавлять к графикам аннотации, трансформировать полученные из запроса данные и использовать переменные нескольких типов.

Сама Grafana запускается как веб приложение, что позволяет развернуть ее один раз, настроить подключение, и просматривать и изменять любимые дашборды в режиме реального времени по сети. В добавок платформа позволяет настроить доступ и аутентификацию.

Все описанные в этом параграфе характеристики представлены для открытого, т.е. “Open Source” решения. Также у компании есть “Enterprise” версия, более оптимизированная для командной работы, и возможность разместить свои дашборды в их облаке, но эти предложения не рассматриваются, т.к. они не актуальны для данной работы.

2.2 Архитектура программного решения

Что-то

2.3 Модификация плагина

Что-то

2.4 Разработка серверной части

Что-то

3 Конструирование дашборда

Заключение

Список использованных источников