АНАЛИЗ ДАННЫХ В ЭКОНОМИКЕ И ФИНАНСАХ

Анализ характеристик и прогнозирование кассовых сборов фильмов

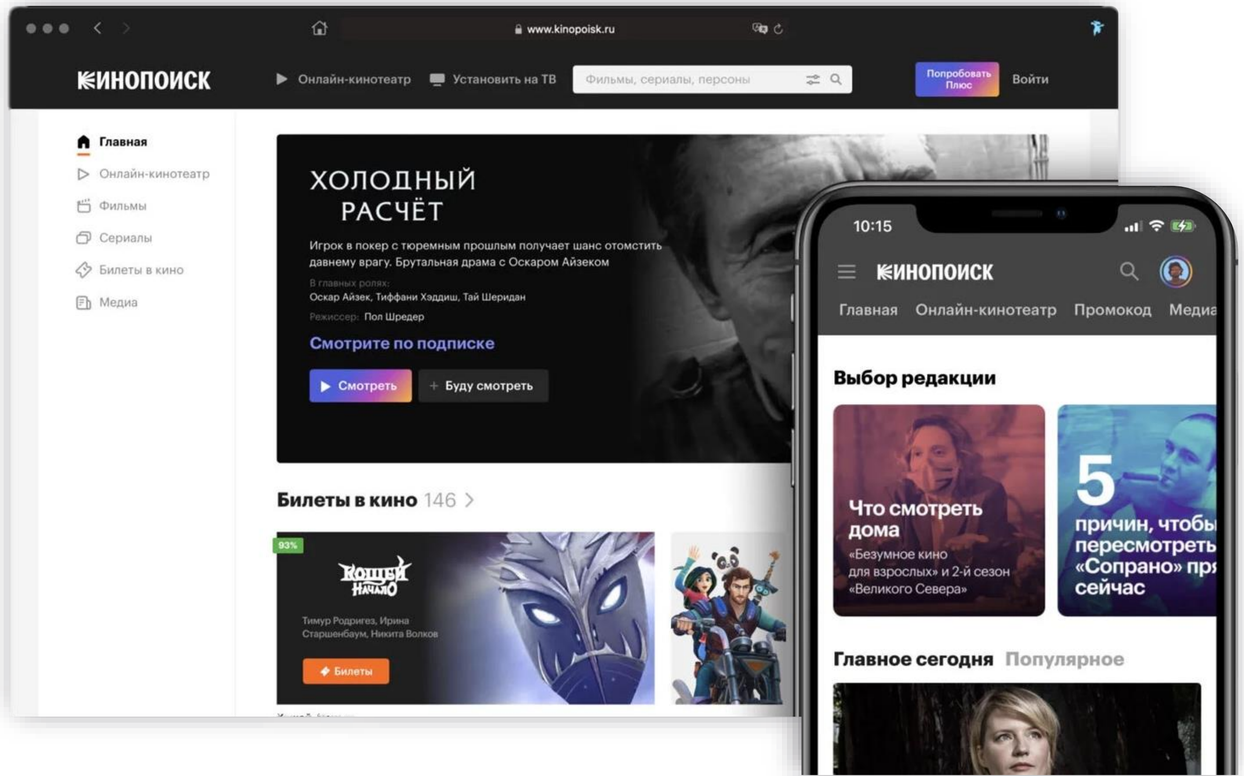
на основе данных сайта **КиноПоиск**

Проект выполнили
Кострова Ксения, 18Э1
Кузнецова Анастасия, 18Э1
Самсонова Алиса, 18Э2
Сучкова Наталья, 18Э1

Обоснование выбора темы

“КиноПоиск” –

крупнейший русскоязычный интернет-сервис о кино, который предоставляет наиболее полную информацию о медиа-материалах различных жанров.



Исследовательские вопросы:

1. Формирование **базы данных**
2. **Исследование премьер** в российском кинопрокате
3. Анализ детерминант и **прогнозирование** кассовых сборов фильмов

Применение:

1. Использование полученной базы данных и дашборда **для других исследований**
2. Использование результатов **для выявления и учета предпочтений** российской публики, возможность построения простейших **рекомендательных алгоритмов**
3. Использование модели **для прогнозирования** кассовых сборов фильмов по известным характеристикам

План работы

01
Сбор данных

Сбор данных при помощи API
Преобразование json файлов
Создание БД и заполнение таблиц на сервере MySQL

02
Разведочный анализ

Базовый разведочный анализ
Заполнение пропущенных значений
Приведение данных к нужным форматам
Приведение валют к текущему курсу, дисконтирование с учетом инфляции
Приведение к Первой нормальной форме
Перенос данных в MySQL
Базовая визуализация

03
Построение дашборда

Написание запросов для выборки данных
Написание кода для графиков (plotly, PIL, stylecloud)
Построение динамического обновления графиков
Сборка дашборда при помощи dash (html, css)

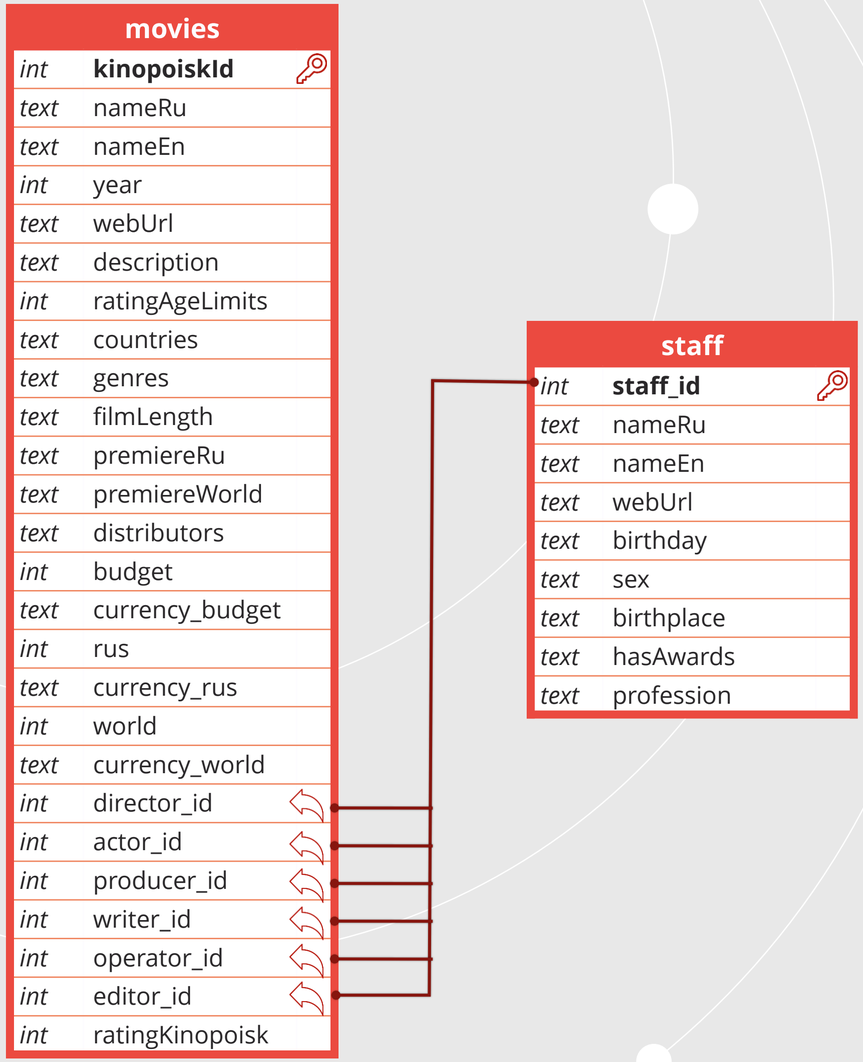
04
Создание и тестирование модели

Линейная регрессия
Гребневая и Lasso-регрессия
Полиномиальная регрессия
Метод ближайших соседей
Деревья решений

СБОР ДАННЫХ

01. СБОР ДАННЫХ

СХЕМА ПЕРВОНАЧАЛЬНОЙ БАЗЫ ДАННЫХ



Первичный сбор данных

Источник данных: https://kinopoiskapiunofficial.tech
(неофициальный API КиноПоиск)

- Ограниченное число запросов (20 q/sec)
- Определенный формат запросов
- Период данных: 2011-2021 гг.
- Среда выполнения кода: Google Colab (бесплатные мощные графические процессоры GPU и TPU, позволяющие достаточно быстро обрабатывать запросы)
- Сбор данных по двум направлениям: формирование первой таблицы с фильмами и их характеристикой, второй - информация об основных создателях
- Формат выгружаемой информации - .json файлы

STEP #1

Формирование БД, таблиц и их наполнение

- Сервер: MySQL сервер
- Preprocessing .json файлов
- Используется: DDL (CREATE, ALTER, DROP), DMT (INSERT)

STEP #2

02. РАЗВЕДОЧНЫЙ АНАЛИЗ

Данные, количество наблюдений:

Таблица 1
movies: 5465 →1749

Таблица 2
staff: 15462

Exploratory analysis:

01 Чистка данных

02 **Обработка пропущенных значений** (часть была найдена и вставлена вручную, пропуски, не представляющие возможность для заполнения, удалены)

03 **Приведение валют к текущему курсу** (бюджеты, кассовые сборы)

04 **Перевод значений** (уникальных признаков: genres - 26, countries - 78, distributors - 86) **в бинарные** для удовлетворения условия Первой нормальной формы таблицы, создание трёх отдельных таблиц

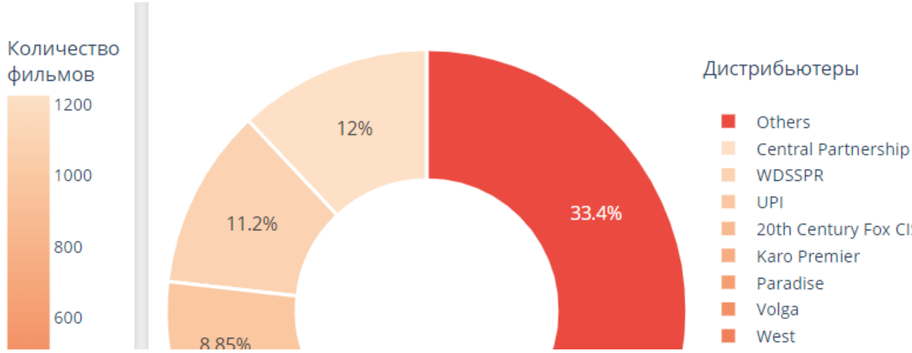
05 **Перенос обновленных данных** в БД DDL (CREATE, ALTER, DROP), DMT (INSERT)

06 **Первичная визуализация** данных для выявления отличительных особенностей данных и возможных зависимостей и закономерностей

СХЕМА ОКОНЧАТЕЛЬНОЙ БАЗЫ ДАННЫХ

03. ПОСТРОЕНИЕ ДАШБОРДА

с динамическим обновлением графиков



step #1
первоначальный / обновленный **запрос SQL** – default value

step #2
функция **построения графика** (plotly, PIL, stylecloud)

step #3
перенос **в дашборд** (dash, html, css)

step #4
при наличии доп. параметров выбора

ПОСТРОЕНИЕ ДАШБОРДА

SQL-ЗАПРОСЫ ДЛЯ ГРАФИКОВ

«Распределение произведенных фильмов по странам»

с выбором временного промежутка

```
CREATE TEMPORARY TABLE temp_table
		SELECT COLUMN_NAME as column_names
		FROM INFORMATION_SCHEMA.columns
		WHERE TABLE_SCHEMA = 'kinopoisk_movies'
		    AND TABLE_NAME = 'countries'
		    AND COLUMN_NAME NOT LIKE 'kinopoiskId';

UPDATE temp_table SET column_names=CONCAT('SUM(',
column_names, ')');

SET session group_concat_max_len = 4096;

CREATE TEMPORARY TABLE temp_table_q
		SELECT GROUP_CONCAT(DISTINCT column_names
SEPARATOR ', ') as query_sum_part
		FROM temp_table;

UPDATE temp_table_q SET query_sum_part = CONCAT('SELECT
year, ', query_sum_part, 'FROM movies_upd mu LEFT JOIN
countries ON mu.kinopoiskId = countries.kinopoiskId GROUP BY
mu.year');
```

```
ALTER TABLE temp_table_q
		ADD COLUMN id INT NOT NULL;

SELECT query_sum_part INTO @sql1 FROM temp_table_q WHERE id =
0;

PREPARE sql_query FROM @sql1;

EXECUTE sql_query;
```

«TOP-5 фильмов по бюджетам и кассовым сборам» (часть)

график без выбора дополнительных параметров

```
SELECT * FROM (SELECT REPLACE (name, ': ', ':<br>') name,
                                                budget_USD_d,
                                                worldBO_USD_d,
                                                rusBO_USD_d
                FROM kinopoisk_movies.movies_upd
                ORDER BY budget_USD_d DESC
                LIMIT 5) table_b
ORDER BY budget_USD_d
```

04. СОЗДАНИЕ МОДЕЛИ

прогнозирование кассовых сборов фильмов в России

	Model	Comment	Metric	Score
0	Linear	with numerous parametres 30/70 split	R^2	0,388008
1	Linear	with parametres *ratingAgeLimits, ratingKinopoisk, budget_USD_d, Adventure, action_movie, WDSSPR* 45/55 split	R^2	0,454929
2	Ridge	auto ridge	R^2	0,454887
3	Ridge	cross validated ridge	R^2	0,447911
4	Polynomial	degree = 2	R^2	0,448229
5	KNN	K-Fold validation, n_neighbors = 5	R^2	0,45397
6	Radius Neighbors	radius = 2, metric = euclidean	R^2	0,376563
7	Desicion Tree	auto	R^2	0,431005
8	**Lasso**	**auto**	**R^2**	**0,454929**

Результаты тестирования различных моделей

Сборы фильмов в России (USD) = 5214720.01 **+** 4983991.29 * *бюджет фильма (USD)*
+ 339673.78 * *жанр приключения* **-** 982025.41 * *рейтинг AgeLimits* **+** 385703.72 * *жанр action*
+ 322562.72 * *дистрибьютор Walt Disney Studio*

БЛАГОДАРИМ ЗА ВНИМАНИЕ!

Будем рады услышать ваши комментарии
и ответить на вопросы.