УДК 004.896

Интеллектуальный анализ данных о посещаемости музеев в разные сезоны года

Акентьев Данила Денисович
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
В данной исследовательской работе исследуется применение методов интеллектуального анализа данных для прогнозирования посещаемости музеев. Актуальность работы обусловлена растущей потребностью в эффективном управлении культурными объектами и оптимизации их деятельности на основе анализа данных. Были рассмотрены современные методы интеллектуального анализа данных, такие как случайный лес, XGBoost, градиентный бустинг и Lasso регрессия, для создания моделей прогнозирования. Для оценки моделей использовались метрики R^2, MAE и MSE. Результаты показали, что все примененные методы продемонстрировали высокую точность предсказаний, при этом модель Lasso регрессии показала наилучшие результаты. Работа представляет собой важный вклад в повышение эффективности управления культурными ресурсами и разработку целевых маркетинговых стратегий для музеев.
Ключевые слова: Интеллектуальный анализ данных, машинное обучение, прогнозирование посещаемости, культурный менеджмент, случайный лес, XGBoost, градиентный бустинг, Lasso регрессия, анализ временных рядов, музейные данные.

Intelligent analysis of museum attendance data in different seasons of the year

Akentev Danila Denisovich
Sholom-Aleichem Priamursky State University
Student

Abstract
This research paper explores the application of data mining methods to predict museum attendance. The relevance of the work is due to the growing need for effective management of cultural sites and optimization of their activities based on data analysis. Modern methods of data mining, such as random forest, XGBoost, gradient boosting and Lasso regression, were considered to create forecasting models. The R^2, MAE and MSE metrics were used to evaluate the models. The results showed that all the applied methods demonstrated high accuracy of predictions, while the Lasso regression model showed the best results. The work represents an important contribution to improving the efficiency of cultural

resource management and the development of targeted marketing strategies for museums.

Keywords: Data mining, machine learning, attendance forecasting, cultural management, random forest, XGBoost, Gradient boosting, Lasso regression, time series analysis, museum data.

1 Введение

1.1 Актуальность

Применение методов интеллектуального анализа данных (ИАД) в различных областях науки также стремительно расширяется, особенно в течение последних десяти лет. ИАД является одним из наиболее важных и влиятельных инструментов в современном мире анализа данных.

ИАД используется в науке для решения широкого спектра задач, включая анализ геномных данных, прогнозирование погоды и климатических изменений, моделирование экосистем, анализ социальных сетей, исследование генетических алгоритмов, и многое другое. Эти методы позволяют обрабатывать и анализировать огромные объемы данных, выявлять закономерности и тенденции, которые не всегда могут быть замечены с помощью традиционных методов анализа.

Наряду с этим, ИАД способствует созданию инновационных решений и открывает новые возможности для научных исследований в различных дисциплинах. Он улучшает способность ученых извлекать знания из данных, предсказывать результаты экспериментов, и вносить существенный вклад в научные открытия и технологические инновации. Таким образом, влияние и потенциал ИАД в науке продолжают расти, что подчеркивает его ключевую роль в современном исследовательском процессе.

Интеллектуальный анализ данных (ИАД) - это процесс извлечения значимых, полезных и интересных знаний из больших объемов данных с использованием различных методов, технологий и инструментов. Он объединяет в себе концепции и подходы из таких областей, как статистика, машинное обучение, искусственный интеллект, анализ данных, визуализация данных и базы данных.

Актуальность данной курсовой работы проявляется в контексте растущей потребности в эффективном управлении культурными объектами, такими как музеи, их посещаемостью и программами развития. В современном мире культурные учреждения сталкиваются с вызовами, связанными с изменяющимися предпочтениями посетителей, конкуренцией за внимание и ресурсы, а также необходимостью демонстрации своей значимости для общества и культурного развития. Анализ посещаемости музеев и выявление сезонных паттернов имеет важное практическое значение для оптимизации работы культурных учреждений, разработки целевых маркетинговых стратегий и создания персонализированных программ для посетителей. Таким образом, данная работа адресует актуальные потребности в области культурного менеджмента и представляет

собой важный вклад в повышение эффективности управления культурными ресурсами.

1.2 Обзор исследований

Исследованиями в данной теме занимались следующие авторы. А.Г.Егоров и Е.Е. Сухова в своей работе провели анализ статистических показателей посещаемости музеев. Обследовались пять разнопрофильных музеев. Зафиксировали динамику роста количества посетителей музеев в 2012 году, а также увеличение посещаемости музеев в обследуемом периоде было сбалансированным [1]. Ю. В. Мигунова рассмотрела в своей статье показатель посещаемости музеев как фактор проявления социально-культурных потребностей населения регионов России [2]. Бренд регионального музея как фактор его посещаемости был рассмотрен И.В.Маракулиной [3]. В работе исследуется взаимосвязь между уровнем осведомленности о брендах региональных музеев и их посещаемостью в контексте маркетинговой стратегии, используя монографический метод, описание, сравнение и обобщение, а также статистические методы и приемы. Проанализировали мотивацию туристов для посещения музеев, включая два измерения мотивации: поиск знаний и рекреационный интерес, и исследуют их влияние на частоту посещений музеев в своей работе такие авторы как Х.Брида, К. Ногаре и Р. Скудери [4]. Д. Тринх и Д. Лам в своей статье рассматривают альтернативный подход к анализу посещаемости культурных объектов и мероприятий, используя стохастические модели поведения потребителей, такие как модель NBD и модель NBD-Дирихле, для прогнозирования поведения посетителей [5]. В. Мартинес-де-Альбенис и А.Вальдивия разработали модель для изучения влияния выставок на количество посетителей музеев и управления им, применяя методы исследования операций, в частности, планирование выставок, как рычаг для повышения воздействия музеев на посетителей [6]. Г. Кафф в своей работе выявил, что влияние дождя на спрос на досуг в помещениях значительно различается в течение дня, и что многие посетители активно изменяют свои планы в течение дня в ответ на дождливую погоду [7].

1.3 Цель исследования

Цель данной статьи заключается в разработке модели прогнозирования посещаемости музеев с использованием методов машинного обучения.

2 Материалы и методы

Для решения поставленной задачи использовались: Анализ временных рядов: Использование методов временных рядов для оценки динамики посещаемости музеев в разные сезоны года и выявления сезонных паттернов. Корреляционный анализ: Построена корреляционная матрица для изучения взаимосвязей между различными переменными. Анализ мультиколлинеарности: для оценки мультиколлинеарности между независимыми переменными использовался метод Variance Inflation Factor

(VIF). Этот анализ помог исключить из регрессионной модели переменные, которые могут вызвать проблемы из-за мультиколлинеарности. Статистический анализ: Использование статистических методов для оценки влияния различных факторов на посещаемость музеев и определения статистической значимости полученных результатов. Визуализация данных: Создание графиков и диаграмм для наглядного представления динамики посещаемости музеев в разные сезоны года и влияния внешних факторов на эту посещаемость. Машинное обучение: Применение методов машинного обучения, таких как случайный лес, xgboost, градиентный бустинг, lasso регрессия для прогнозирования посещаемости музеев на основе исторических данных и факторов внешней среды.

Случайный лес (Random forest) — алгоритм машинного обучения, предложенный Лео Брейманом и Адель Катлер, заключающийся в использовании комитета (ансамбля) решающих деревьев. Алгоритм сочетает в себе две основные идеи: метод бэггинга Бреймана, и метод случайных подпространств, предложенный Тин Кам. Алгоритм применяется для задач классификации, регрессии и кластеризации. Основная идея заключается в использовании большого ансамбля решающих деревьев, каждое из которых само по себе даёт очень невысокое качество классификации, но за счёт их большого количества результат получается хорошим.

XGBoost (eXtreme Gradient Boosting) – это алгоритм машинного обучения, который основан на методе градиентного бустинга. Он был разработан и предложен Чень Лионгом в 2016 году и с тех пор стал одним из наиболее популярных алгоритмов в области анализа данных и машинного обучения. Основная идея XGBoost заключается в построении ансамбля слабых моделей, обычно деревьев решений, последовательно, при этом каждая новая модель исправляет ошибки предыдущих. Однако, в отличие от классического градиентного бустинга, XGBoost использует дополнительные техники для ускорения обучения и повышения качества модели.

Градиентный бустинг (Gradient Boosting) — ¬это алгоритм машинного обучения, который также основан на ансамбле моделей, но в отличие от случайного леса, в котором каждое дерево строится независимо друг от друга, в градиентном бустинге новые модели добавляются к ансамблю последовательно и улучшаются на основе ошибок предыдущих моделей. Основная идея градиентного бустинга заключается в том, чтобы на каждом шаге строить новую модель, которая корректирует ошибки предыдущих моделей. Это достигается путем минимизации функции потерь, которая измеряет разницу между предсказанными и реальными значениями целевой переменной

Лассо-регрессия — это метод линейной регрессии, который добавляет штраф к модели, чтобы снизить переобучение и получить более простую модель. Он основан на минимизации функции потерь, включая регуляризационный член, который штрафует модель за слишком большие значения коэффициентов. Этот метод особенно полезен в задачах отбора признаков, где он может автоматически устанавливать некоторые

коэффициенты признаков в ноль, тем самым выполняя отбор наиболее информативных признаков. Таким образом, лассо-регрессия помогает создать более интерпретируемую модель за счет сокращения размерности и улучшения обобщающей способности.

Оценка моделей прогнозирования измеряется в метриках R^2, MAE и MSE. R^2 (R-квадрат) — это коэффициент детерминации в статистике. Он измеряет пропорцию изменчивости зависимой переменной, объясненную (или объяснимую) моделью. Значение R^2 находится между 0 и 1, где 1 означает идеальное соответствие модели данным, а 0 означает отсутствие такого соответствия. MAE (Mean Absolute Error) — это средняя абсолютная ошибка. Он измеряет среднее абсолютное отклонение между прогнозами модели и фактическими значениями. Чем ниже значение MAE, тем лучше модель. MSE (Mean Squared Error) — это среднеквадратичная ошибка. Он измеряет среднее значение квадратов отклонений между прогнозами модели и фактическими значениями. Важно отметить, что MSE увеличивает вес больших ошибок по сравнению с MAE. Как и в случае MAE, чем меньше значение MSE, тем лучше модель.

3 Обзор программного обеспечения

Перед тем как приступить к интеллектуальному анализу следует рассмотреть среду выполнения, а именно Google Colab.

Google Colab представляет собой мощное облачное средство для анализа данных с использованием языка программирования Python. Это идеальное решение как для новичков, так и для опытных специалистов, поскольку предоставляет доступ к широкому спектру инструментов машинного обучения и анализа данных. С помощью Google Colab пользователи могут проводить анализ данных, создавать визуализации и запускать машинное обучение без необходимости устанавливать и настраивать среду на своем локальном компьютере (Рис.1).

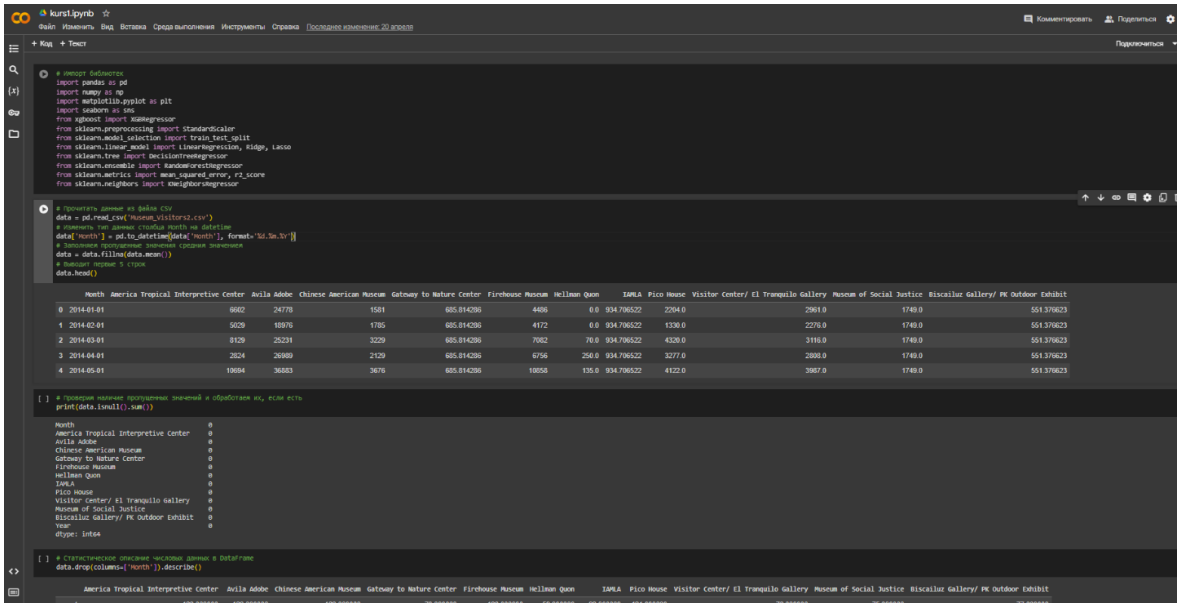


Рисунок 1. Google Colab

4 Обзор набора данных

Данные были взяты с сайта Data.Gov. Этот набор данных содержит информацию о посещаемости различных музейных объектов в разные месяцы с января 2014 года по март 2024 года. Данные включают количество посетителей, посещающих каждый месяц, в различных музейных объектах, таких как "America Tropical Interpretive Center", "Avila Adobe", "Chinese American Museum" и другие. Информация была собрана от соответствующих музейных организаций или учреждений и, вероятно, включает данные, собранные с помощью системы учета посещений или анкетирования посетителей (Рис.2).

index	Month	America Tropical Interpretive Center	Avila Adobe	Chinese American Museum	Gateway to Nature Center	Firehouse Museum	Hellman Quon	IAMLA	Pico House	Visitor Center/ El Tranquilo Gallery	Museum of Social Justice	Biscailuz Gallery/ PK Outdoor Exhibit
0	Jan 2014	6602	24778	1581	NaN	4486	0.0	NaN	2204.0	2961.0	NaN	NaN
1	Feb 2014	5029	18976	1785	NaN	4172	0.0	NaN	1330.0	2276.0	NaN	NaN
2	Mar 2014	8129	25231	3229	NaN	7082	70.0	NaN	4320.0	3116.0	NaN	NaN
3	Apr 2014	2824	26989	2129	NaN	6756	250.0	NaN	3277.0	2808.0	NaN	NaN
4	May 2014	10694	36883	3676	NaN	10858	135.0	NaN	4122.0	3987.0	NaN	NaN
5	Jun 2014	11036	29487	2121	NaN	5751	255.0	NaN	355.0	3133.0	NaN	NaN
6	Jul 2014	13490	32378	2239	NaN	5406	120.0	NaN	3375.0	3527.0	NaN	NaN
7	Aug 2014	9139	37680	1769	NaN	8619	250.0	NaN	1550.0	0.0	NaN	NaN
8	Sep 2014	5661	28473	1073	NaN	61192	1145.0	NaN	1335.0	0.0	NaN	NaN
9	Oct 2014	7356	27995	1979	NaN	6488	550.0	NaN	1518.0	0.0	NaN	NaN
10	Nov 2014	9773	25691	2404	NaN	4189	410.0	NaN	7769.0	0.0	NaN	NaN
11	Dec 2014	7184	18754	1319	NaN	4339	565.0	NaN	1140.0	0.0	NaN	NaN
12	Jan 2015	6250	20438	1823	NaN	3858	75.0	NaN	200.0	2082.0	NaN	NaN
13	Feb 2015	5907	15578	1558	NaN	3742	160.0	NaN	1075.0	2751.0	NaN	NaN
14	Mar 2015	9884	21297	2336	NaN	5390	325.0	NaN	3445.0	2748.0	NaN	NaN
15	Apr 2015	7254	26670	3057	NaN	7000	2000.0	NaN	3292.0	3408.0	NaN	NaN
16	May 2015	13207	34383	4009	NaN	12528	475.0	NaN	5042.0	3785.0	NaN	NaN
17	Jul 2015	11072	41242	3057	NaN	6111	100.0	NaN	4225.0	4233.0	NaN	NaN
18	Jun 2015	11102	30569	2544	NaN	5377	200.0	NaN	4552.0	3444.0	NaN	NaN
19	Aug 2015	12096	30700	2415	NaN	5383	50.0	NaN	5974.0	3606.0	NaN	NaN
20	Sep 2015	6608	20967	1398	NaN	5746	125.0	NaN	700.0	3323.0	NaN	NaN
21	Oct 2015	12524	29764	2237	NaN	8882	750.0	NaN	4158.0	3209.0	NaN	NaN
22	Nov 2015	6677	24483	2850	NaN	6848	950.0	NaN	9312.0	13750.0	NaN	NaN
23	Dec 2015	5967	21426	2075	NaN	4502	120.0	NaN	170.0	3367.0	NaN	NaN
24	Jan 2016	6587	19659	2150	NaN	4377	50.0	NaN	250.0	2940.0	NaN	NaN

Рисунок 2. Набор данных

По данному скриншоту можно сделать вывод, что данные нуждаются в предварительной обработке.

Переводим столбец Month в формат даты с указанием первого числа каждого месяца. После чего обработаем пропущенные значения путем заполнения данных ячеек средним показателем по столбцу.

Теперь данные обработаны и готовы к работе (Рис.3).

index	Month	America Tropical Interpretive Center	Avila Adobe	Chinese American Museum	Gateway to Nature Center	Firehouse Museum	Hellman Quon	IAMLA	Pico House	Visitor Center/ El Tranquilo Gallery	Museum of Social Justice	Biscailuz Gallery/ PK Outdoor Exhibit
0	2014-01-01 00:00:00	6602	24778	1581	685.8142857142857	4486	0.0	934.7065217391304	2204.0	2961.0	1749.0	551.3766233766233
1	2014-02-01 00:00:00	5029	18976	1785	685.8142857142857	4172	0.0	934.7065217391304	1330.0	2276.0	1749.0	551.3766233766233
2	2014-03-01 00:00:00	8129	25231	3229	685.8142857142857	7082	70.0	934.7065217391304	4320.0	3116.0	1749.0	551.3766233766233
3	2014-04-01 00:00:00	2824	26989	2129	685.8142857142857	6756	250.0	934.7065217391304	3277.0	2808.0	1749.0	551.3766233766233
4	2014-05-01 00:00:00	10694	36883	3676	685.8142857142857	10858	135.0	934.7065217391304	4122.0	3987.0	1749.0	551.3766233766233
5	2014-06-01 00:00:00	11036	29487	2121	685.8142857142857	5751	255.0	934.7065217391304	355.0	3133.0	1749.0	551.3766233766233
6	2014-07-01 00:00:00	13490	32378	2239	685.8142857142857	5406	120.0	934.7065217391304	3375.0	3527.0	1749.0	551.3766233766233
7	2014-08-01 00:00:00	9139	37680	1769	685.8142857142857	8619	250.0	934.7065217391304	1550.0	0.0	1749.0	551.3766233766233
8	2014-09-01 00:00:00	5661	28473	1073	685.8142857142857	61192	1145.0	934.7065217391304	1335.0	0.0	1749.0	551.3766233766233
9	2014-10-01 00:00:00	7356	27995	1979	685.8142857142857	6488	550.0	934.7065217391304	1518.0	0.0	1749.0	551.3766233766233
10	2014-11-01 00:00:00	9773	25691	2404	685.8142857142857	4189	410.0	934.7065217391304	7769.0	0.0	1749.0	551.3766233766233
11	2014-12-01 00:00:00	7184	18754	1319	685.8142857142857	4339	565.0	934.7065217391304	1140.0	0.0	1749.0	551.3766233766233
12	2015-01-01 00:00:00	6250	20438	1823	685.8142857142857	3858	75.0	934.7065217391304	200.0	2082.0	1749.0	551.3766233766233
13	2015-02-01 00:00:00	5907	15578	1558	685.8142857142857	3742	160.0	934.7065217391304	1075.0	2751.0	1749.0	551.3766233766233
14	2015-03-01 00:00:00	9884	21297	2336	685.8142857142857	5390	325.0	934.7065217391304	3445.0	2748.0	1749.0	551.3766233766233
15	2015-04-01 00:00:00	7254	26670	3057	685.8142857142857	7000	2000.0	934.7065217391304	3292.0	3408.0	1749.0	551.3766233766233
16	2015-05-01 00:00:00	13207	34383	4009	685.8142857142857	12528	475.0	934.7065217391304	5042.0	3785.0	1749.0	551.3766233766233
17	2015-07-01 00:00:00	11072	41242	3057	685.8142857142857	6111	100.0	934.7065217391304	4225.0	4233.0	1749.0	551.3766233766233
18	2015-06-01 00:00:00	11102	30569	2544	685.8142857142857	5377	200.0	934.7065217391304	4552.0	3444.0	1749.0	551.3766233766233
19	2015-08-01 00:00:00	12096	30700	2415	685.8142857142857	5383	50.0	934.7065217391304	5974.0	3606.0	1749.0	551.3766233766233
20	2015-09-01 00:00:00	6608	20967	1398	685.8142857142857	5746	125.0	934.7065217391304	700.0	3323.0	1749.0	551.3766233766233
21	2015-10-01 00:00:00	12524	29764	2237	685.8142857142857	8882	750.0	934.7065217391304	4158.0	3209.0	1749.0	551.3766233766233
22	2015-11-01 00:00:00	6677	24483	2850	685.8142857142857	6848	950.0	934.7065217391304	9312.0	13750.0	1749.0	551.3766233766233
23	2015-12-01 00:00:00	5967	21426	2075	685.8142857142857	4502	120.0	934.7065217391304	170.0	3367.0	1749.0	551.3766233766233
24	2016-01-01											

Рисунок 3. Обработанные данные

5 Результаты и обсуждения

Начнем исследование с импорта библиотек. Это можно сделать при помощи кода, представленного ниже (Рис.4).

```
# Импорт библиотек
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from xgboost import XGBRegressor
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.linear_model import Lasso
from sklearn.ensemble import GradientBoostingRegressor
import xgboost as xgb
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

Рисунок 4. Импорт нужных библиотек

Далее загрузим данные, представленные в формате csv файла. Так же определим столбец Month в формат даты и заполним все пропущенные значения средним. Сразу выведем первые 5 строк для просмотра датасета. Воспользуемся кодом, представленным ниже (Рис. 5.1-5.2).

```
# Прочитать данные из файла CSV
data = pd.read_csv('Museum_Visitors2.csv')
# Преобразование столбца 'Month' в формат DateTime
data['Month'] = pd.to_datetime(data['Month'], format='%d.%m.%Y')
# Заполняем пропущенные значения средним значением
data = data.fillna(data.mean())
# Выводит первые 5 строк
data.head()
```

Рисунок 5.1. Код для чтения, преобразования и заполнения данных

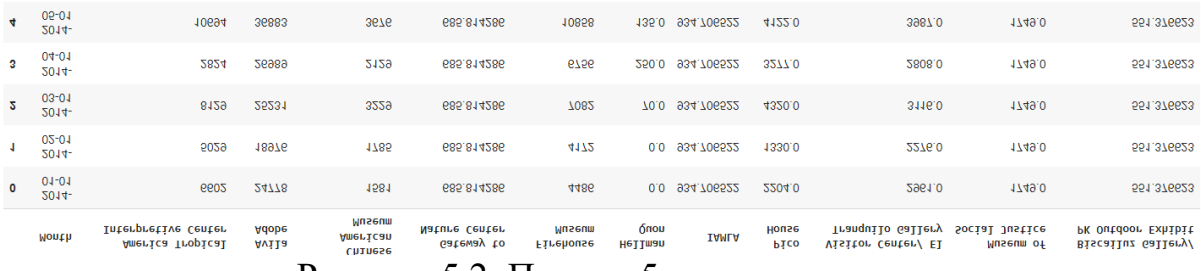


Рисунок 5.2. Первые 5 строк датасета

Проверим наличие пропущенных значений, чтобы убедиться в правильности выполнения предыдущего кода. Иначе в будущем они могут создать проблемы (Рис.6).

```
# Проверим наличие пропущенных значений
print(data.isnull().sum())
```

```
Month                                  0
America Tropical Interpretive Center   0
Avila Adobe                            0
Chinese American Museum                0
Gateway to Nature Center               0
Firehouse Museum                       0
Hellman Quon                           0
IAMLA                                  0
Pico House                             0
Visitor Center/ El Tranquilo Gallery   0
Museum of Social Justice               0
Biscailuz Gallery/ PK Outdoor Exhibit  0
dtype: int64
```

Рисунок 6. Проверка пропущенных значений

Выведем статическое описание числовых данных, чтобы получить обзорную информацию о распределении значений, меры центральной тенденции (например, среднее), разброса данных (например, стандартное отклонение) и основные квантили. Это помогает понять основные характеристики данных, выявить аномалии или выбросы, а также принять решения о дальнейшей обработке или анализе данных (Рис.7.1-7.2).

```
# Статистическое описание числовых данных в DataFrame
data.drop(columns=['Month']).describe()
```

Рисунок 7.1. Код для статистического описания числовых данных

	America Tropical Interpretive Center	Avila Adobe	Chinese American Museum	Gateway to Nature Center	Firehouse Museum	Hellman Quon	IAMLA	Pico House	Visitor Center/ El Tranquilo Gallery	Museum of Social Justice	Biscailuz Gallery/ PK Outdoor Exhibit
count	123.000000	123.000000	123.000000	123.000000	123.000000	123.000000	123.000000	123.000000	123.000000	123.000000	123.000000
mean	4446.886179	16452.333333	1960.918699	685.814286	4373.910569	155.762712	934.706522	922.446281	1520.076923	1749.000000	551.376623
std	3299.629714	9729.715926	1355.732381	903.181102	5680.493726	236.314612	736.261453	1621.384138	1614.680021	1206.162483	569.762275
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	2040.500000	10129.500000	1065.000000	0.000000	2659.000000	0.000000	660.000000	0.000000	0.000000	1321.000000	0.000000
50%	4081.000000	17378.000000	2078.000000	685.814286	3930.000000	155.762712	934.706522	220.000000	1520.076923	1749.000000	551.376623
75%	6476.000000	23269.500000	2575.000000	685.814286	5214.500000	155.762712	1073.500000	1077.500000	2106.000000	1931.000000	726.000000
max	13490.000000	41242.000000	7702.000000	6000.000000	61192.000000	2000.000000	5565.000000	9312.000000	13750.000000	10740.000000	3801.000000

Рисунок 7.2 Результат

Построим матрицы диаграмм рассеяния для всех числовых переменных в DataFrame. Они позволяют визуализировать взаимосвязи между переменными. Это помогает исследовать структуру данных, выявлять потенциальные зависимости и корреляции, а также обнаруживать выбросы или необычные образцы, что может быть полезно при исследовательском анализе данных и выборе подходящих моделей машинного обучения (Рис.8.1-8.2).

```
# Построение матрицы диаграмм рассеяния для всех числовых переменных в DataFrame
sns.pairplot(data)
plt.show()
```

Рисунок 8.1. Код для построения матрицы диаграмм рассеяния

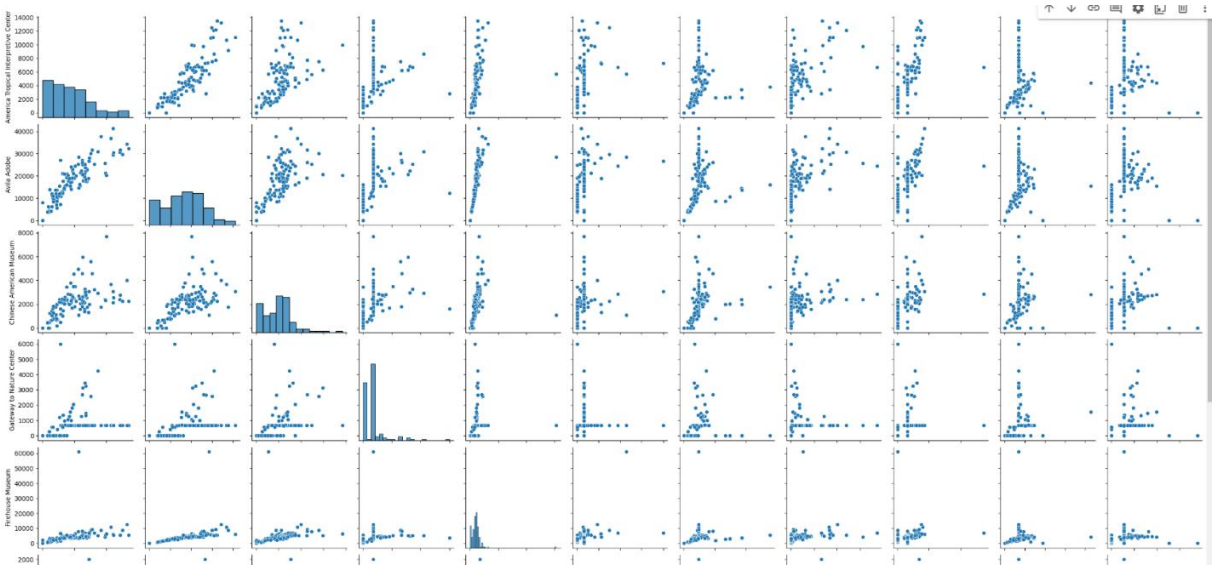
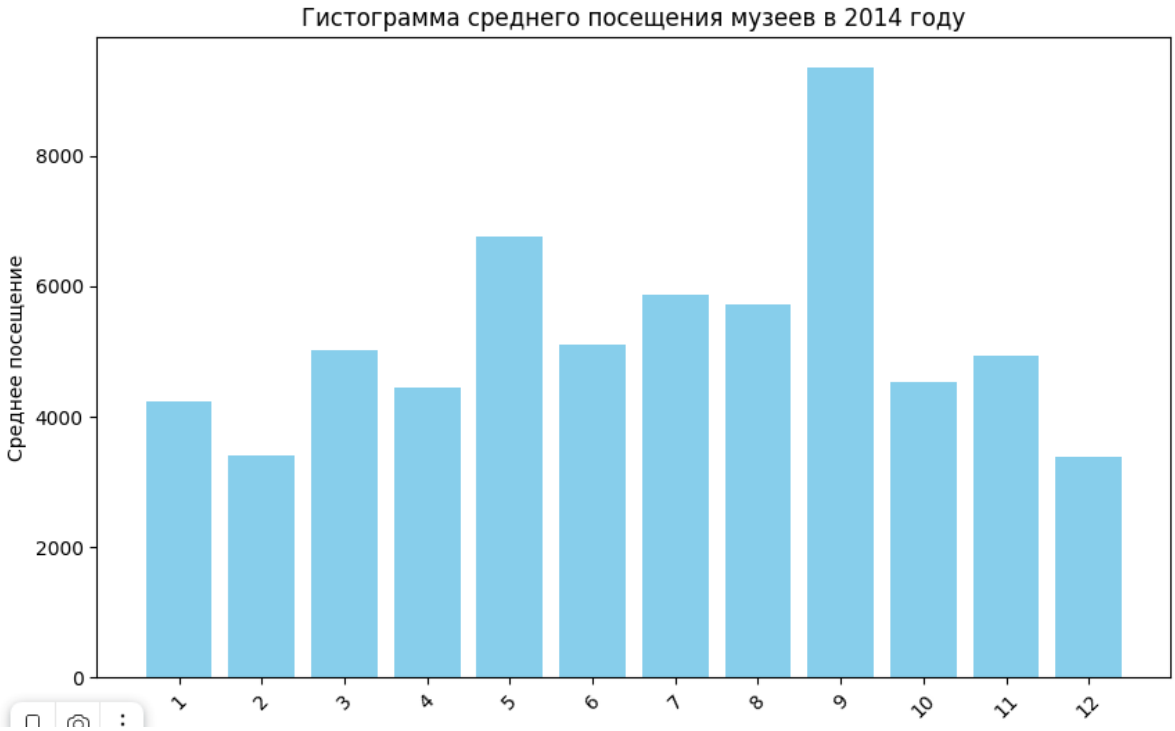


Рисунок 8.2. Результат

Далее сгруппируем данные по месяцам и по годам, а также вычислим среднее посещение для того, чтобы построить гистограммы (Рис.9.1-9.2).

```
# Группировка данных по месяцам
mean_attendance_per_month = data.groupby('Month').mean()
# Вычисление среднего значения
mean_per_row = mean_attendance_per_month.mean(axis=1)
# Группировка данных по годам
grouped_data = mean_per_row.groupby(mean_per_row.index.year)
# Построение гистограммы для каждого года
for year, year_data in grouped_data:
    plt.figure(figsize=(10, 6))
    plt.bar(year_data.index.month, year_data.values, color='skyblue')
    plt.title(f'Гистограмма среднего посещения музеев в {year} году')
    plt.xlabel('Месяц')
    plt.ylabel('Среднее посещение')
    plt.xticks(range(1, 13), rotation=45)
    plt.show()
```

Рисунок 9.1. Код для построения гистограмм



9.2. Пример результат

Теперь выполним анализ временных рядов. Анализ временных рядов важен для изучения изменений в данных во времени и выявления паттернов, трендов, сезонности или аномалий. Этот анализ позволяет прогнозировать будущие значения, выявлять факторы, влияющие на данные, и принимать обоснованные решения на основе прошлых тенденций. Для начала требуется создать копию дата фрейма и установить столбец Month в качестве индекса (Рис. 10).

```
#Создание копии DataFrame
data_copy = data.copy()
#Установка индекса в копии DataFrame
data_copy.set_index('Month', inplace=True)
```

Рисунок 10. Создание копии и установка индекса

Далее сгруппируем по месяцам и годам, после чего построим график временного ряда (Рис.11.1-11.2).

```
#Группировка данных по месяцам
mean_attendance_per_month = data_copy.groupby(data_copy.index).mean()
#Группировка данных по годам
grouped_data = mean_attendance_per_month.groupby(mean_attendance_per_month.index.year)
# Построение графика временного ряда
for year, year_data in grouped_data:
    plt.figure(figsize=(10, 6))
    plt.plot(year_data.index.month, year_data.mean(axis=1), marker='o', linestyle='-')
    plt.title(f'Среднее посещение музеев в {year} году')
    plt.xlabel('Месяц')
    plt.ylabel('Среднее посещение')
    plt.xticks(range(1, 13), rotation=45)
    plt.grid(True)
    plt.show()
```

Рисунок 11.1 Код для построения графика временного ряда.

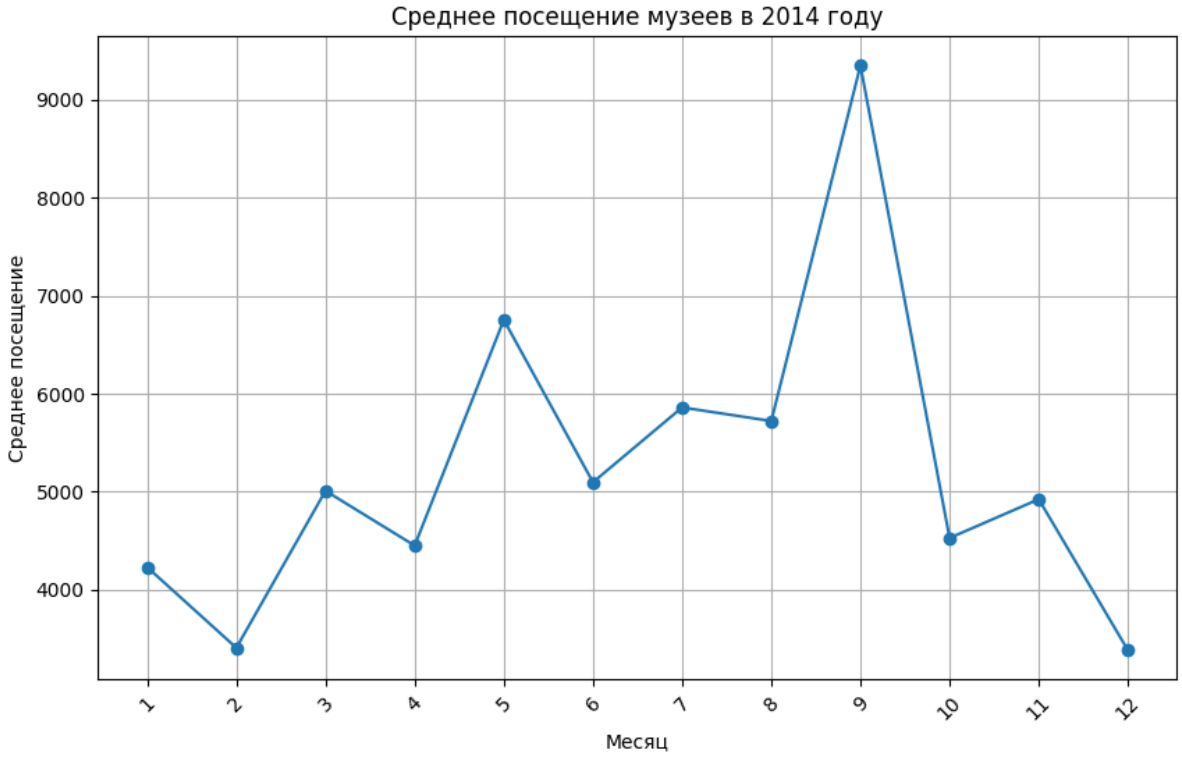


Рисунок 11.2. Пример результата

В результате видим, что в 2014 г. max посещений был в сентябре, а min это февраль и декабрь. Также в 2015 – 2019 г. max посещений был в мае, а min в феврале. Из чего можно сделать вывод, что за последние 10 лет пиком по количеству посещений был месяц май.

Далее проведем корреляционный анализ. Корреляционный анализ необходим для определения степени взаимосвязи между двумя или более переменными. Этот метод позволяет выявить, существует ли прямая или обратная зависимость между переменными, что помогает понять, как изменения в одной переменной связаны с изменениями в другой. Такой анализ часто используется для исследования взаимосвязи между факторами и результатами, что помогает принимать более обоснованные решения в

различных областях, включая экономику, медицину, социологию и многие другие.

Вычислим коэффициенты корреляции и визуализируем матрицу корреляции (Рис.12.1-12.2).

```
# Вычисление коэффициентов корреляции
correlation_matrix = data.corr()
# Визуализация матрицы корреляции
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, cmap='coolwarm', annot=True, fmt=".2f", linewidths=.5)
plt.title('Матрица корреляции')
plt.show()
```

Рисунок 12.1. Код матрицы корреляции

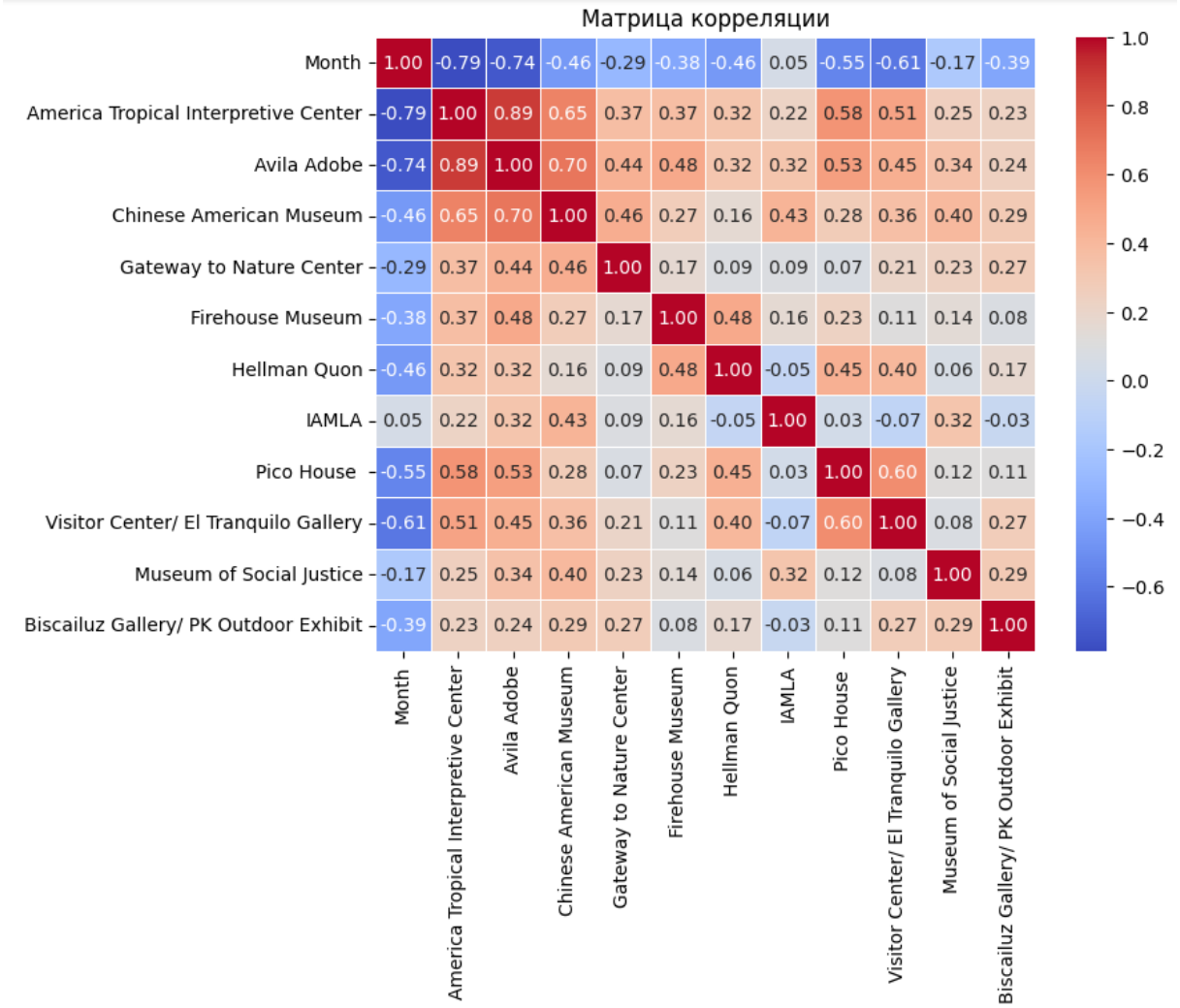


Рисунок 12.2. Матрица корреляции

Теперь проведем оценку степени корреляции (VIF) со столбцом Month. Оценка степени корреляции со столбцом "Month" позволяет выявить, насколько другие переменные изменяются в зависимости от месяца. Это важно для исследования сезонных паттернов или трендов, а также для

выявления возможных влияний временных факторов на данные, что может помочь в принятии решений или разработке стратегий, учитывающих сезонные колебания (Рис.13).

VIF расшифровывается как Variance Inflation Factor, что означает "фактор мультиколлинеарности" в статистике. Это статистический показатель, используемый для оценки степени мультиколлинеарности в регрессионном анализе. Он позволяет оценить, насколько сильно между собой коррелируют независимые переменные в модели регрессии. Высокие значения VIF указывают на наличие мультиколлинеарности, что может привести к нестабильным оценкам коэффициентов и затруднить интерпретацию результатов регрессионного анализа. Обычно считается, что VIF больше 10 является признаком серьезной мультиколлинеарности.

```
# Оценка степени корреляции
pearson_corr = correlation_matrix['Month'].abs().sort_values(ascending=False)
print("Коэффициенты корреляции с Month:")
print(pearson_corr)
```

```
Коэффициенты корреляции с Month:
Month                                  1.000000
America Tropical Interpretive Center   0.786970
Avila Adobe                            0.738083
Visitor Center/ El Tranquilo Gallery   0.606100
Pico House                             0.547830
Hellman Quon                           0.457314
Chinese American Museum                0.456907
Biscailuz Gallery/ PK Outdoor Exhibit  0.391800
Firehouse Museum                       0.383533
Gateway to Nature Center               0.287819
Museum of Social Justice               0.166226
IAMLA                                  0.048127
Name: Month, dtype: float64
```

Рисунок 13. Коэффициенты корреляции с Month

Можем увидеть, что все столбцы не сильно коррелируют с данным столбцом, значит ничто не помешает написать модели машинного обучения для предсказания.

Следующим действием проведем анализ мультиколлинеарности. Анализ мультиколлинеарности необходим для выявления проблемы, когда два или более предикторных переменных в модели линейной регрессии сильно коррелируют между собой. Это важно, так как мультиколлинеарность может исказить оценки коэффициентов и статистическую значимость модели, делая ее менее интерпретируемой и менее надежной. Этот анализ помогает исследователям принимать решения о включении или исключении переменных из модели для улучшения ее качества и предсказательной силы.

Для начала создадим копию дата фрейма и удалим столбец Month. Изначально задавали данному столбцу формат даты. Но VIF измеряется для столбцов числового формата (Рис.14).

```
# Вычисляем VIF для каждого признака
vif_data = pd.DataFrame()
vif_data["Feature"] = data_copy_1.columns
vif_data["VIF"] = [variance_inflation_factor(data_copy_1.values, i) for i in range(data_copy_1.shape[1])]

print("VIF для каждого признака:")
print(vif_data)
```

```
VIF для каждого признака:
                                Feature        VIF
0    America Tropical Interpretive Center  15.983837
1                            Avila Adobe  25.650703
2                 Chinese American Museum   8.082715
3                Gateway to Nature Center   2.255909
4                        Firehouse Museum   2.757275
5                            Hellman Quon   2.404676
6                                   IAMLA   3.570674
7                             Pico House    2.786657
8    Visitor Center/ El Tranquilo Gallery   3.630041
9                Museum of Social Justice   3.902658
10  Biscailuz Gallery/ PK Outdoor Exhibit   2.405183
```

Рисунок 14. VIF

С помощью данного анализа можно увидеть, что первые 3 строки выдают слишком высокие результаты. Поэтому удалим данные признаки для улучшения в будущем нашей модели (Рис.15).

```
data_copy_1 = data.drop(columns = ['Month',
                                   'Avila Adobe',
                                   'America Tropical Interpretive Center',
                                   'Chinese American Museum'] )
```

Рисунок 15. Удаление признаков с высоким значением VIF

Далее снова вычисляем VIF и смотрим на результаты. (Рис.16)

```
# Вычисляем VIF для каждого признака
vif_data = pd.DataFrame()
vif_data["Feature"] = data_copy_1.columns
vif_data["VIF"] = [variance_inflation_factor(data_copy_1.values, i) for i in range(data_copy_1.shape[1])]

print("VIF для каждого признака:")
print(vif_data)
```

```
VIF для каждого признака:
                              Feature       VIF
0              Gateway to Nature Center  1.845438
1                      Firehouse Museum  2.226920
2                          Hellman Quon  2.384082
3                                 IAMLA  2.565357
4                           Pico House   2.327447
5   Visitor Center/ El Tranquilo Gallery  3.181444
6              Museum of Social Justice  3.541612
7  Biscailuz Gallery/ PK Outdoor Exhibit  2.343904
```

Рисунок 16. Коэффициенты мультиколлинеарности после удаление признаков

Теперь данные полностью готовы к созданию моделей прогнозирования.

Реализация случайного леса (Random Forest)

Перед началом построения модели создадим копию дата фрейма и удалим столбцы, которые превышали норму по VIF (Рис.17).

```
data_copy_2 = data.copy()
data_copy_2 = data.drop(columns = ['Month',
                                   'Avila Adobe',
                                   'America Tropical Interpretive Center',
                                   'Chinese American Museum'] )
```

Рисунок 17. Копия, удаление столбцов дата фрейма

После чего разделим данные на 2 группы test и train. Сделаем это с помощью кода, представленного ниже (Рис.18).

```
Model = data_copy_2
Musei = data_copy_2['Museum of Social Justice']

# Разделим данные на тренировочную и тестовую выборки в соотношении 80/20
Model_train, Model_test, Musei_train, Musei_test = train_test_split(Model, Musei, test_size=0.2, random_state=42)
```

Рисунок 18. Разделение выборок на тестовые и тренировочные

Далее создадим и обучим модели на тренировочной выборке. После чего мы сделаем предсказание и оценим данную модель. Также визуализируем 3 величины оценки данной модели (Рис. 19).

```
# Создание и обучение модели
random_forest = RandomForestRegressor(random_state=42)
random_forest.fit(Model_train, Musei_train)
# Предсказание на тестовых данных
Musei_pred_rf = random_forest.predict(Model_test)
# Оценка модели
mae_rf = mean_absolute_error(Musei_test, Musei_pred_rf)
mse_rf = mean_squared_error(Musei_test, Musei_pred_rf)
r2_rf = r2_score(Musei_test, Musei_pred_rf)

print("Случайный лес:")
print("MAE:", mae_rf)
print("MSE:", mse_rf)
print("R^2:", r2_rf)
```

```
Случайный лес:
MAE: 35.9312
MSE: 8623.321327999993
R^2: 0.9840993964062976
```

Рисунок 19. Random Forest

Модель случайного леса демонстрирует высокую точность предсказаний, так как значение R^2 близко к 1, а MAE и MSE относительно низки. Это говорит о том, что модель хорошо соотносится с данными и способна точно предсказывать целевую переменную.

Реализация XGBoost (Extreme Gradient Boosting)

Для построения модели XGBoost, достаточно создать и обучить модель, а также провести предсказание и оценку данной модели. Выполним это с помощью кода, представленного ниже. И вновь визуализируем эти данные (Рис.20).

```
# Создание и обучение модели
xgb_reg = xgb.XGBRegressor(random_state=42)
xgb_reg.fit(Model_train, Musei_train)

# Предсказание на тестовых данных
Musei_pred_xgb = xgb_reg.predict(Model_test)

# Оценка модели
mae_xgb = mean_absolute_error(Musei_test, Musei_pred_xgb)
mse_xgb = mean_squared_error(Musei_test, Musei_pred_xgb)
r2_xgb = r2_score(Musei_test, Musei_pred_xgb)

print("XGBoost:")
print("MAE:", mae_xgb)
print("MSE:", mse_xgb)
print("R^2:", r2_xgb)
```

```
XGBoost:
MAE: 21.07928508002311
MSE: 1983.8422317852185
R^2: 0.9963419791840948
```

Рисунок 20. XGBoost

Модель XGBoost демонстрирует еще более высокую точность предсказаний, чем модель случайного леса, так как значения MAE, MSE и R^2 близки к идеальным. Это говорит о том, что модель XGBoost отлично соотносится с данными и способна очень точно предсказывать целевую переменную.

Реализация градиентного бустинга (Gradient Boosting)

Данный метод реализуем с помощью определенной библиотеки, которую мы импортировали в самом начале. Так же, как и в предыдущем пункте нужно создать, обучить, предсказать и оценить модель (Рис.21).

```
# Создание и обучение модели
gradient_boosting = GradientBoostingRegressor(random_state=42)
gradient_boosting.fit(Model_train, Musei_train)

# Предсказание на тестовых данных
Musei_pred_gb = gradient_boosting.predict(Model_test)

# Оценка модели
mae_gb = mean_absolute_error(Musei_test, Musei_pred_gb)
mse_gb = mean_squared_error(Musei_test, Musei_pred_gb)
r2_gb = r2_score(Musei_test, Musei_pred_gb)

print("Градиентный бустинг:")
print("MAE:", mae_gb)
print("MSE:", mse_gb)
print("R^2:", r2_gb)
```

```
Градиентный бустинг:
MAE: 20.672786643479302
MSE: 2935.8956739018035
R^2: 0.994586481063671
```

Рисунок 2. Gradient Boosting

Модель градиентного бустинга также демонстрирует высокую точность предсказаний, похожую на модель XGBoost. Значения MAE, MSE и R^2 близки к идеальным, что указывает на то, что модель градиентного бустинга хорошо соотносится с данными и способна очень точно предсказывать целевую переменную.

Реализация Lasso регрессии (Lasso regression)

Следующий метод реализуется так же, как и предыдущие. Создадим и обучим Lasso модель с помощью тренировочной выборки. После чего предскажем на тестовых данных. И в конце оценим данную модель (Рис.22).

```
# Создание и обучение Lasso модели
lasso_reg = Lasso(alpha=0.1)  # Выбор коэффициента регуляризации (alpha)
lasso_reg.fit(Model_train, Musei_train)

# Предсказание на тестовых данных
Musei_pred_lasso = lasso_reg.predict(Model_test)

# Оценка модели
mae_lasso = mean_absolute_error(Musei_test, Musei_pred_lasso)
mse_lasso = mean_squared_error(Musei_test, Musei_pred_lasso)
r2_lasso = r2_score(Musei_test, Musei_pred_lasso)

print("Lasso регрессия:")
print("MAE:", mae_lasso)
print("MSE:", mse_lasso)
print("R^2:", r2_lasso)
```

```
Lasso регрессия:
MAE: 0.014152527713251024
MSE: 0.0005307819945650165
R^2: 0.9999999990212873
```

Рисунок 22. Lasso regression

Модель Lasso регрессии демонстрирует выдающуюся точность предсказаний. Значения MAE, MSE и R^2 очень близки к идеальным, что указывает на то, что модель Lasso регрессии отлично соотносится с данными и способна очень точно предсказывать целевую переменную.

Выводы

Результат исследования можно представить в таблице 1.

Таблица 1. Сравнение оценок различных моделей классификации

Модель	MAE	MSE	R^2
Random Forest	35.931	8623.321	0.984
XGBoost	21.079	1983.842	0.996
Gradient Boosting	20.673	2935.896	0.995
Lasso regression	0.014	0.00053	0.999

Можно сделать следующие выводы:

Random Forest демонстрирует высокую точность предсказаний с умеренными значениями MAE и MSE, а также высоким значением R^2, что указывает на хорошее соответствие модели данным.

XGBoost и Gradient Boosting также показывают очень высокую точность среди всех моделей, имея низкие значения MAE и MSE и высокий коэффициент детерминации R^2, что свидетельствует о хорошей способности моделей объяснять изменчивость зависимой переменной.

Lasso регрессия демонстрирует наивысшую точность среди всех моделей, имея очень низкие значения MAE и MSE, а также практически идеальное значение R^2, что указывает на то, что эта модель имеет наименьшее среднее абсолютное отклонение и квадраты отклонений от фактических значений, и хорошо соотносится с данными.

Все представленные модели демонстрируют высокую точность и эффективность в предсказании значений зависимой переменной для данного датасета. Однако модель Lasso регрессии выделяется на фоне остальных моделей как наиболее точная и наименее склонная к ошибкам. Таким образом, использование различных методов машинного обучения для решения данной задачи может быть рекомендовано, однако для наилучшего результата стоит предпочесть модель Lasso регрессии.

Ноутбук в Google Colab URL: https://colab.research.google.com/drive/1fBNbFTluaUfoncbvztySO7thZhe1YoSZ?usp=sharing

Библиографический список

1. Егорова А.Г., Сухова Е.Е. Динамика посещаемости Смоленских музеев в 2009-2012 годах: Анализ статистических данных // Социальные трансформации. 2013. № 23. С. 39-49.
2. Мигунова Ю.В. Показатель посещаемости музеев как фактор проявления социально-культурных потребностей населения регионов России // Общество: социология, психология, педагогика. 2019. № 1(57). С. 13-15.
3. Маракулина И.В. Бренд регионального музея как фактор его посещаемости // Наука Красноярья. 2020. Т.9. № 1. С. 146-158.
4. Brida J.G., Nogare C.D., Scuderi R. Frequency of museum attendance: motivation matters // Journal of Cultural Economics. 2015. Т. 40. С. 261-283.
5. Trinh G., Lam D. Understanding the attendance at cultural venues and events with stochastic preference models // Journal of Business Research. 2016. Т. 69. № 9. С. 3538-3544.
6. Martinez-de-Albeniz V., Valdivia A. Measuring and Exploiting the Impact of Exhibition Scheduling on Museum Attendance // Manufacturing & Service Operations Management. 2018. Т. 21. № 4. С. 713-948.
7. Cuffe H. Rain and museum attendance: Are daily data fine enough? // Journal of Cultural Economics. 2017. Т. 42. С. 213-241.