Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г. В. Плеханова»

Высшая школа кибертехнологий, математики и статистики
Кафедра информатики
Направление 38.03.05 Бизнес-информатика
Профиль "Цифровая трансформация бизнеса"

Выпускная квалификационная работа

по программе профессиональной переподготовки «Анализ данных и машинное обучение в среде Python»

на тему «Прогнозирование индекса счастья на основе методов машинного обучения»

Выполнила студентка группы: 15.11Д-БИЦТ09/21б
2 курс, факультет: ВШКМИС
Мамонтова Татьяна Юрьевна

_____
(подпись)

Проверила:

ст. преп. Савинова Виктория Михайловна

_____ _____
(оценка) *(подпись)*

_____
(дата)

Москва
2023

1. Постановка задачи исследования

1.1.Понимание бизнеса

Исследование направлено на то, чтобы повысить эффективность показателя - индекс счастья, благодаря которому странам легче определить сферу, требующую дополнительного финансирования, чтобы повысить уровень жизни и благополучие их населения. Соответственно, необходимо:

1. Изучить все доступные факторы, входящие в расчет индекса счастья;
2. С помощью статистического анализа и моделей машинного обучения определить, какие из этих факторов оказывают наибольшее влияние на уровень счастья населения, а какие имеют меньший, но потенциально значимый вклад;
3. Предоставить результаты исследования политикам, ученым и другим заинтересованным сторонам для использования при разработке мер, направленных на повышение благосостояния населения.

Ожидаемые преимущества - данный анализ обеспечит более точное понимание структуры индекса счастья и может быть использован для принятия обоснованных управленческих решений и разработки стратегий повышения благосостояния населения.

Критерий успеха - выявление совокупности факторов, определяющих показатель счастья населения.

1.2.Доступные ресурсы

Для успешной реализации проекта необходимы следующие категории специалистов: аналитик данных, бизнес-аналитик, руководитель проекта.

Заказчик располагает всем необходимым оборудованием для проведения анализа данных.

1.3.Риски

1. Несоблюдение сроков проекта;
2. Риск неплатежеспособности заказчика;
3. Риск нехватки и неполноты данных;
4. Наличие выбросов и искажённых данных может повлиять на качество прогнозов;
5. Риск несоответствия полученных результатов требованиям заказчика.

1.4.Ограничения

Ограничение по срокам выполнения проекта — 2 месяца.

Анализ проводится только на данных 2018 года, без использования многолетних рядов.

Небольшой объём данных (156 наблюдений, один год).

Исследуются только регрессионные алгоритмы классического машинного обучения.

1.5.Цели исследования данных

1. Проведение разведочного анализа данных, включающего: проверку наличия пропусков и дубликатов, формирование таблицы описательной статистики, построение гистограмм и boxplot для числовых признаков, визуализацию распределения целевой переменной (индекса счастья), а также графическое сравнение стран с наибольшим и наименьшим уровнем счастья. Дополнительно проводится проверка наличия выбросов с помощью метода межквартильного размаха до и после процедуры винзоризации. На заключительном этапе разведочного анализа строится корреляционная матрица для оценки взаимосвязей между признаками и целевой переменной.

2. Решение задачи прогнозирования индекса счастья проводится с использованием методов регрессионного анализа. Рассматриваются несколько моделей машинного обучения: линейная регрессия, ridge-регрессия, полиномиальная регрессия, метод ближайших соседей (KNN), а также метод случайного леса.

1.6.Критерии успешности изучения данных

Метрики оценки точности и качества построенных моделей.

Для моделей регрессии:

Коэффициент детерминации (R^2) - отражает долю объяснённой вариации целевой переменной.

Средняя абсолютная процентная ошибка (MAPE) - показывает относительную точность предсказаний в процентах.

Средняя абсолютная ошибка (MAE) - характеризует среднюю величину отклонения прогнозов от реальных значений.

Среднеквадратичная ошибка (RMSE) - учитывает квадратичное отклонение прогнозов.

Границы значений метрик: R2 должен быть больше либо равен 0.8, MAPE не более 10%, MAE и RMSE чем меньше, тем лучше.

2. Начальное изучение данных

2.1.Сбор данных

Внутренние данные: 'Overall rank' – (Общий ранг: Список рангов разных стран от 1 до 156), 'Country or region' – (Страна или регион: Список названий разных стран), 'Score' – (Индекс: список показателей - индекса счастья в разных странах), 'GDP per capita' – (Показатель ВВП на душу населения в разных странах), 'Social support' – (Социальная поддержка разных стран), 'Healthy life expectancy' – (Ожидаемая продолжительность здоровой жизни в разных странах), 'Freedom to make life choices' – (Свобода делать жизненный выбор: оценка восприятия свободы в разных странах), 'Generosity' – (Щедрость: качество быть добрым и щедрым, оценка разных стран), 'Perceptions of corruption' – (Оценка восприятия коррупции в разных странах).

(https://www.kaggle.com/datasets/sougatapramanick/happiness-index-2018-2019)

Для дальнейших прогнозов были выбраны данные за 2018 год.

Внешние данные – не требуются

Дополнительные данные – не требуются

2.2.Описание данных

Объем данных – 18,3 Кбайт

Типы, виды данных и схемы кодирования

Наименование	Тип данных	Вид данных	Схема кодирования
Overall rank	int	дискретный	-
Country or region	string	дискретный	Целое число
GDP per capita	float	непрерывный	-
Social support	float	непрерывный	-
Healthy life expectancy	float	непрерывный	-
Freedom to make life choices	float	непрерывный	-
Generosity	float	непрерывный	-
Perceptions of corruption	float	непрерывный	-

Формат данных – файл csv, разделитель – “,”.

2.3.Исследование данных

В исходном наборе данных отсутствуют пропуски и дубликаты, что подтверждает корректность структуры и полноту информации.

Проверка пропущенных значений

Overall rank 0
Country or region 0
Score 0
GDP per capita 0
Social support 0
Healthy life expectancy 0
Freedom to make life choices 0
Generosity 0
Perceptions of corruption 0

Пропущенные значения отсутствуют.

Дубликаты отсутствуют.

Columns: [Country or region, Score, GDP per capita, Social support, Healthy life expectancy, Freedom to make life choices, Generosity, Perceptions of corruption]

Index: []

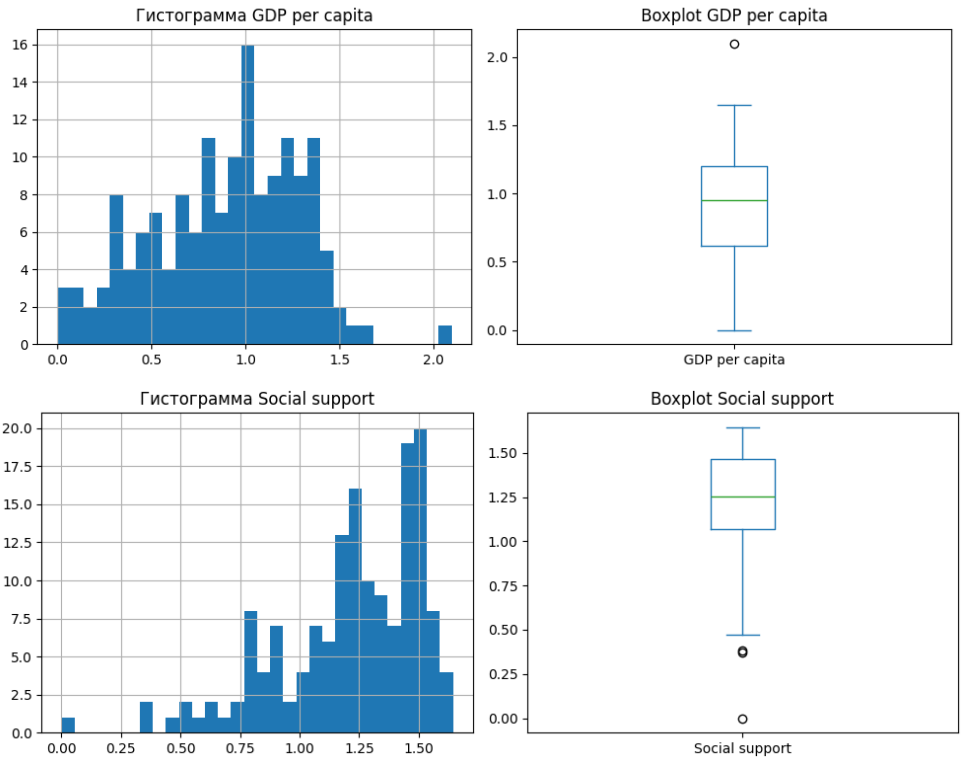
Было выполнено удаление одного столбца Overall rank ввиду его не информативности

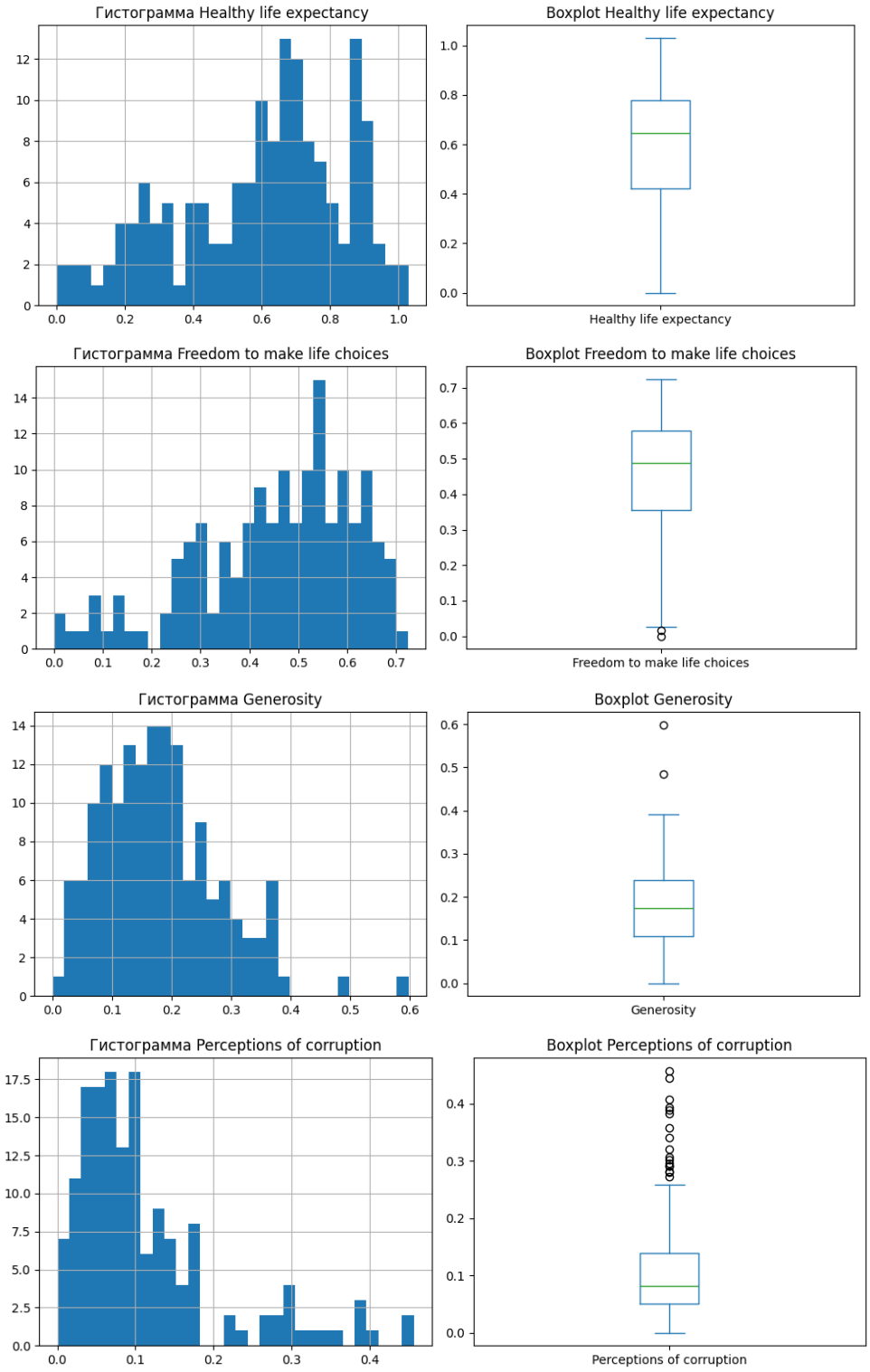
Построение описательной статистики:

	Score	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption
count	156.000000	156.000000	156.000000	156.000000	156.000000	156.000000	156.000000
mean	5.375917	0.891449	1.213237	0.597346	0.454506	0.181006	0.112449
std	1.119506	0.391921	0.302372	0.247579	0.162424	0.098471	0.096343
min	2.905000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	4.453750	0.616250	1.066750	0.422250	0.356000	0.109500	0.051000
50%	5.378000	0.949500	1.255000	0.644000	0.487000	0.174000	0.082000
75%	6.168500	1.197750	1.463000	0.777250	0.578500	0.239000	0.139000
max	7.632000	2.096000	1.644000	1.030000	0.724000	0.598000	0.457000

Анализ описательной статистики показал, что данные требуют дополнительной подготовки перед построением прогнозных моделей. Для некоторых факторов, например Perceptions of corruption, наблюдается значительный разброс значений: стандартное отклонение (std) заметно отличается от среднего (mean), а также видны нетипичные отклонения по квартилям. Это свидетельствует о наличии выбросов, которые необходимо обработать соответствующими методами, чтобы минимизировать их влияние на результаты анализа.

Описание гистограмм распределения и boxplot





Анализ гистограмм и диаграмм размаха показал, что распределение большинства признаков является несимметричным и скошенным вправо.

- Для GDP per capita, Social support и Healthy life expectancy основная масса значений сосредоточена в среднем диапазоне, но присутствуют страны с высокими значениями, образующие «длинный хвост».
- Freedom to make life choices имеет более компактное распределение, но с выбросами в нижней части.

- Generosity и особенно Perceptions of corruption демонстрируют значительное количество выбросов и аномальных значений. На boxplot это проявляется в виде множества точек, выходящих за пределы «усов».

Соответственно, эти особенности необходимо учитывать и обрабатывать при подготовке данных к построению моделей прогнозирования.

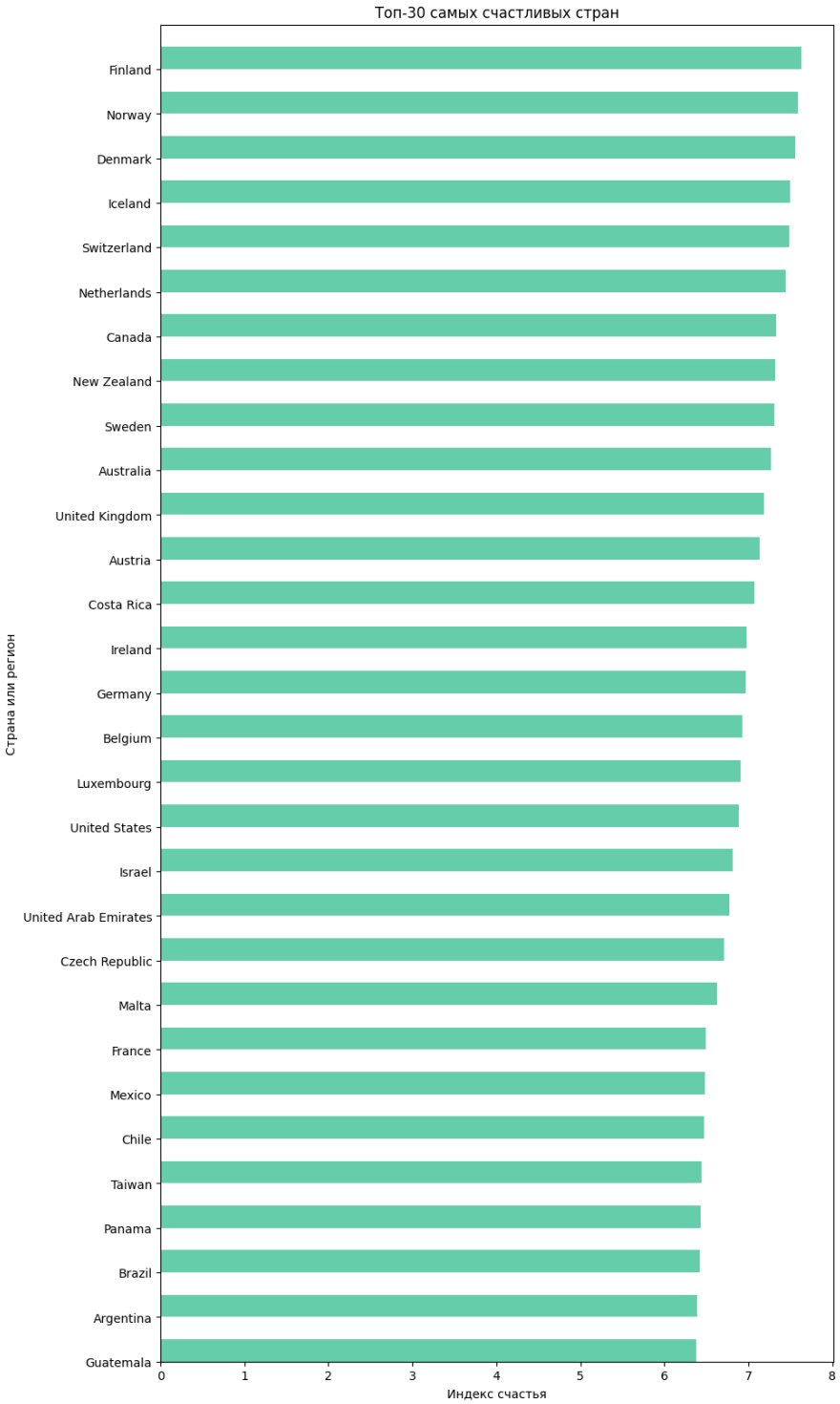
Для получения общего представления о распределении индекса счастья и сравнительного анализа между странами были построены дополнительные графики.

1.

На данном графике представлено распределение по целевой переменной (Score – показатель индекса счастья). График позволяет наглядно отразить, жители скольких стран в той или иной степени удовлетворены своим уровнем жизни. Соответственно, у большинства регионов этот показатель колеблется примерно от 4 до 6. Это говорит о том, что в целом большая часть населения планеты не отличается высоким уровнем благосостояния ввиду следующих факторов, влияющих на индекс счастья: ВВП на душу населения, социальная поддержка, ожидаемая продолжительность здоровой жизни, свобода в принятии решений, щедрость и восприятие коррупции.

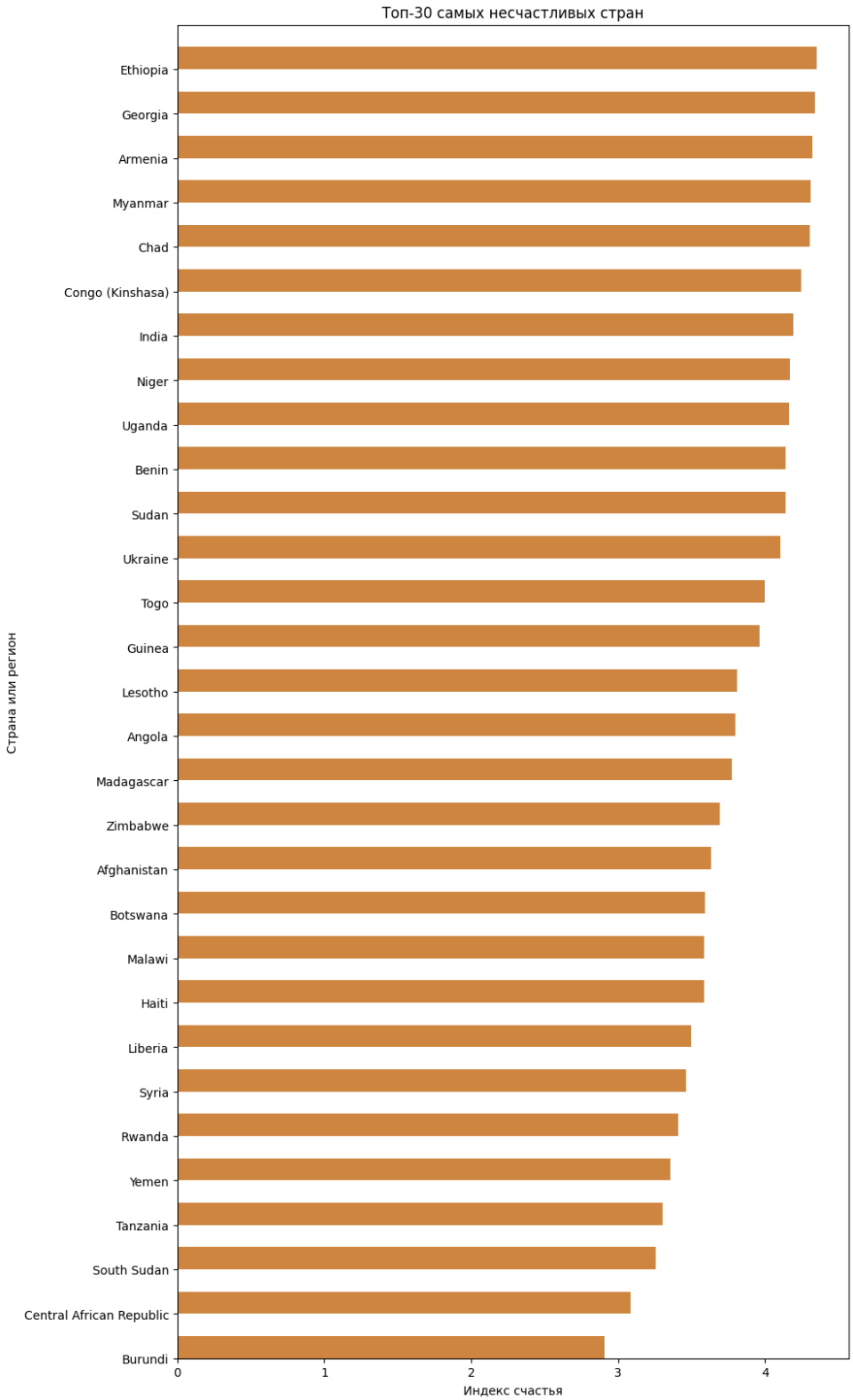
Важно отметить, что количество стран, которых можно считать «несчастливыми», опираясь на данный индекс, значительно меньше, чем удовлетворенных своим положением.

2.



На графике представлены первые 30 стран с наивысшим значением индекса счастья. Из распределения видно, что индекс счастья не связан напрямую с климатическими условиями: в первой двадцатке практически отсутствуют страны с тёплым климатом и обилием солнечных дней, за исключением Израиля, Коста-Рики и Объединённых Арабских Эмиратов. В то же время среди лидеров находится Исландия, где преобладают дождливые и пасмурные дни. Для большинства стран, демонстрирующих высокий уровень счастья, характерна политическая стабильность, длительное отсутствие военных конфликтов, а также высокий уровень жизни в сочетании с развитой системой социальной поддержки.

3.



На графике представлены страны с наименьшими значениями индекса счастья. Большинство из них расположены в Африке или соседних регионах Азии. Для этих стран характерны низкий уровень экономического развития, высокая вовлеченность в военные и политические конфликты, слабая система социальной поддержки и низкая продолжительность жизни населения.

Последнюю позицию в рейтинге занимает Бурунди. По данным международных организаций, около 60% населения страны живёт за чертой бедности, а средняя ожидаемая продолжительность жизни не превышает 40 лет. Эти социально-экономические и

демографические факторы во многом объясняют крайне низкий показатель индекса счастья.

В целом, анализ показывает, что для стран с наименьшим уровнем счастья решающими факторами оказываются экономическая нестабильность, политические кризисы, вооружённые конфликты и ограниченный доступ к базовым услугам здравоохранения и образования.

3. Подготовка данных

После анализа распределений и диаграмм размаха было проведено дополнительное исследование выбросов с использованием метода межквартильного размаха (IQR). Данный метод был выбран, поскольку он является универсальным и не требует предположения о нормальности распределения данных. В отличие от стандартных статистических критериев, метод IQR позволяет выявлять выбросы и в асимметричных распределениях, которые характерны для большинства социально-экономических показателей в рассматриваемом наборе данных.

Проверка проводилась по каждому независимому фактору, включённому в анализ и потенциально влияющему на целевую переменную. Ниже приведено количество выбросов, обнаруженных по методу IQR.

Выбросы в данных:
GDP per capita: 1 выбросов из 156 строк
Social support: 3 выбросов из 156 строк
Healthy life expectancy: 0 выбросов из 156 строк
Freedom to make life choices: 2 выбросов из 156 строк
Generosity: 2 выбросов из 156 строк
Perceptions of corruption: 17 выбросов из 156 строк

С учётом того, что объём выборки составляет всего 156 наблюдений, удаление строк с выбросами приведет к ещё большему сокращению данных и может ухудшить качество обучения моделей. Поэтому было принято решение не удалять выбросы, а обработать их с помощью метода винсоризации.

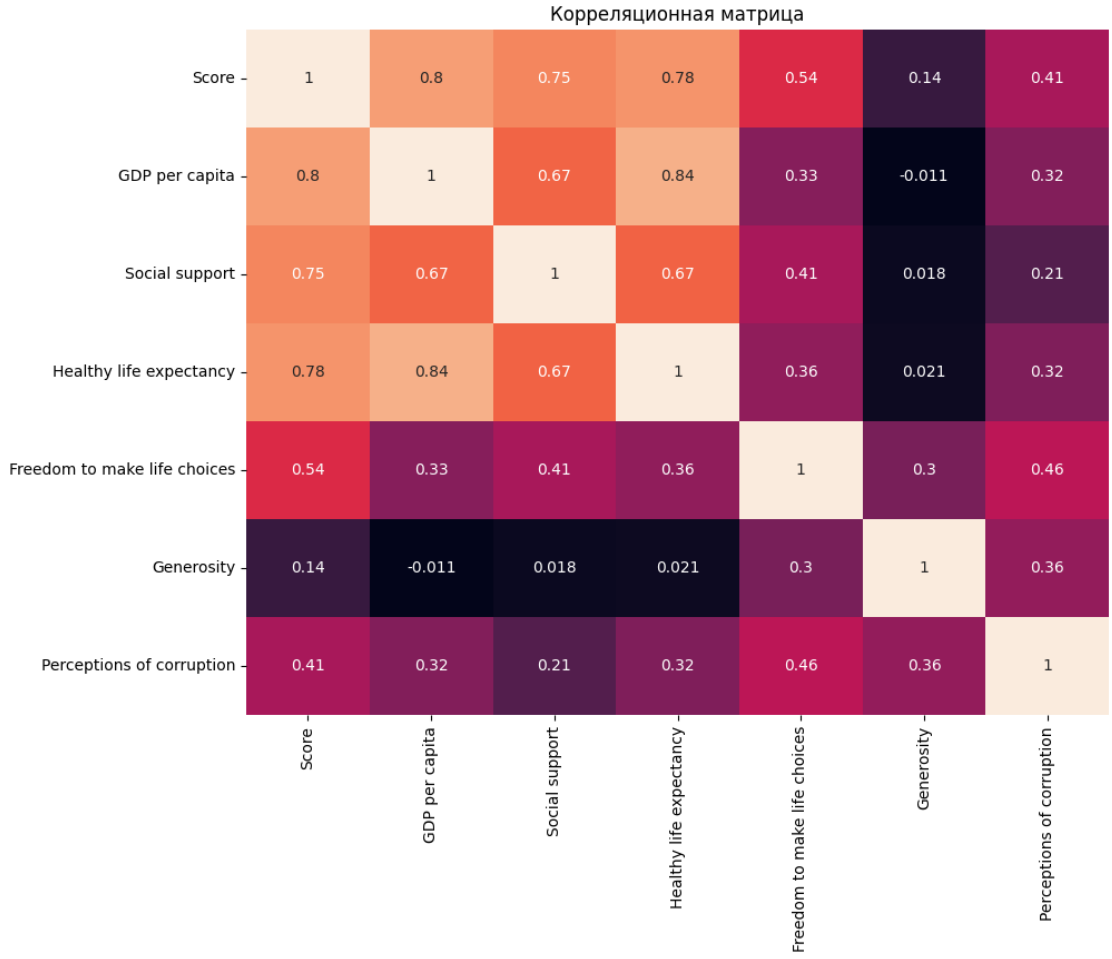
Винсоризация заключается в том, что экстремальные значения признаков заменяются на ближайшие допустимые границы, рассчитанные по межквартильному размаху. Такой подход позволяет сгладить влияние аномально больших или малых значений, при этом сохраняя все строки в выборке. В отличие от удаления выбросов, винсоризация сохраняет структуру данных и обеспечивает моделям больше информации для обучения, что особенно важно при небольшом размере выборки.

Выбросы в данных после применения метода винсоризации:
GDP per capita: 0 выбросов из 156 строк
Social support: 0 выбросов из 156 строк
Healthy life expectancy: 0 выбросов из 156 строк
Freedom to make life choices: 0 выбросов из 156 строк
Generosity: 0 выбросов из 156 строк

Perceptions of corruption: 0 выбросов из 156 строк

Построение корреляционной матрицы

В данную матрицу не включается фактор – Overall rank ввиду его не информативности.



Построена корреляционная матрица между целевой переменной (Score) и факторами, влияющими на индекс счастья. Наиболее сильная корреляция наблюдается между Score и такими признаками, как GDP per capita (0,80), Social support (0,75) и Healthy life expectancy (0,78). Также заметна связь с Freedom to make life choices (0,54). Остальные показатели имеют более слабую линейную зависимость, однако были сохранены для дальнейшего анализа. Это связано с тем, что низкая корреляция не исключает возможного нелинейного или косвенного влияния признака на уровень счастья.

Следует отметить наличие мультиколлинеарности, в частности высокой корреляции между GDP per capita и Healthy life expectancy (0,84). В традиционной линейной регрессии это могло бы исказить интерпретацию коэффициентов. Однако в исследовании применяются не только линейные методы, но и модели, устойчивые к мультиколлинеарности (например, случайный лес и метод ближайших соседей). Поскольку основной целью является построение точных прогностических моделей, а не только интерпретация отдельных коэффициентов, для дальнейшего анализа были оставлены все факторы.

4. Моделирование

Для построения прогностических моделей данные были разделены на обучающую (80%) и тестовую (20%) выборки. Для каждой модели реализована отдельная функция, которая включает этап масштабирования признаков (где это необходимо), обучение модели и оценку качества на тестовой выборке с использованием метрик R^2, MAPE, MAE и RMSE.

Использованные модели:

- Линейная регрессия - базовый метод, оценивающий линейную зависимость между независимыми признаками и индексом счастья. Масштабирование признаков применялось для корректной работы модели с коэффициентами.
- Ridge-регрессия - модификация линейной регрессии с L2-регуляризацией, позволяющая избежать переобучения и уменьшить влияние мультиколлинеарности.
- Полиномиальная регрессия - расширение линейной модели за счёт включения полиномиальных признаков второго порядка для учета возможных нелинейных зависимостей.
- Случайный лес - ансамблевый метод на основе решающих деревьев. Подбор гиперпараметров осуществлялся с помощью GridSearchCV, чтобы оптимизировать выбор глубины деревьев и их количество в ансамбле.
- Метод k-ближайших соседей (KNN) - модель, основанная на близости объектов в пространстве признаков. Для корректной работы применялось масштабирование, а оптимальные параметры подбирались с использованием GridSearchCV.

Результаты обучения и тестирования моделей.

-- Линейная регрессия --
R^2: 0.746348343135455
MAPE: 0.08133474022026782
MAE: 0.44881744666592843
RMSE: 0.5426502183186621

-- Ridge регрессия --
R^2: 0.7472254225332461
MAPE: 0.08114190040490009
MAE: 0.447977473170761
RMSE: 0.5417112150427003

-- Полиномиальная регрессия (deg=2) --
R^2: 0.7943147087411
MAPE: 0.06840083831950754
MAE: 0.391389011546317
RMSE: 0.4886554516746313

-- Случайный лес --
R^2: 0.7780649629108694
MAPE: 0.07033698629482743

MAE: 0.3947710932893872
RMSE: 0.5075911779850374

-- KNN --
R^2: 0.8101650724372917
MAPE: 0.07123911458728521
MAE: 0.3944709165331514
RMSE: 0.46944983502689025

5. Оценка результатов

Результаты обучения и тестирования показали, что наилучшие значения метрик продемонстрировала модель k-ближайших соседей (KNN): коэффициент детерминации R^2 составил 0.81, означая объяснение примерно 81% вариации индекса счастья, а средняя абсолютная процентная ошибка (MAPE) — около 7,1%, что соответствует критерию успешности (≤ 10%). При этом значения MAE (0.39) и RMSE (0.47) также оказались минимальными среди всех рассмотренных моделей, подтверждая высокое качество прогнозов.

Метрики качества удовлетворяют заданным. Принятие модели в эксплуатацию зависит от заказчика.

Полиномиальная регрессия и метод случайного леса также показали высокие результаты (R^2 = 0.79 и R^2 = 0.78 соответственно), однако уступили KNN по точности. Базовые модели линейной и Ridge-регрессии продемонстрировали более низкие значения R^2 (~0.75), подтверждая ограниченность их применимости в данной задаче.

Таким образом, KNN оказался наиболее эффективным на данных 2018 года. Учёт локальных и нелинейных закономерностей обеспечивает преимущество перед классическими линейными моделями.

6. Внедрение

Рекомендуется внедрить наилучшую модель, а именно метод k-ближайших соседей, показавший лучшие результаты по всем метрикам качества. В обучение были включены все факторы, определяющие индекс счастья. Такой выбор обусловлен необходимостью учитывать как основные предикторы, так и показатели с менее выраженной корреляцией, которые тем не менее могут вносить вклад в прогноз в совокупности с другими переменными.

Применение данной модели позволяет комплексно оценивать, какие факторы оказывают наибольшее влияние на уровень благосостояния населения. Результаты анализа могут быть использованы для выявления направлений, требующих особого внимания со стороны государственных структур или общественных организаций. Данное исследование поможет принимать более обоснованные управленческие решения и эффективнее распределять ресурсы.

Приложение 1. Код программы Python по проведенному анализу

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
```

```
import seaborn as sns

from sklearn.linear_model import LinearRegression, Ridge
from sklearn.metrics import r2_score, mean_absolute_percentage_error, mean_absolute_error, mean_squared_error
from sklearn.neighbors import KNeighborsRegressor
from sklearn.preprocessing import LabelEncoder, StandardScaler, PolynomialFeatures
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.pipeline import Pipeline

# Импорт данных
df = pd.read_csv(r'/content/2018 (2).csv')

# Смотрим общую информацию по данным и первые строки
print("Информация о датафрейме:")
df.info()

print("Первые строки датафрейма:")
df.head()

# Удаляем колонку Overall rank, так как это просто индекс рейтинга,
# который напрямую зависит от Score и будет мешать моделям
df.drop(["Overall rank"], axis=1, inplace = True)

#проверка пропущенных значений
def data_gapes(data):
    gapes = data.isnull().sum()
    return gapes
print(data_gapes(df),  "\n")

#выявление дубликатов
def data_dublicates(data):
    dublicats = data[data.duplicated()]
    return dublicats
print (data_dublicates(df),  "\n")

#Считаем базовые статистики по всем числовым признакам
print("Описательная статистика:\n")
df.describe()

# Анализ распределения данных (гистограммы + boxplot)
numeric_columns = df.select_dtypes(include=[np.number]).columns.tolist()
# исключим страну
if "Country or region" in numeric_columns:
    numeric_columns.remove("Country or region")
```

```
for col in numeric_columns:
    plt.figure(figsize=(10, 4))
    plt.subplot(1, 2, 1)
    df[col].hist(bins=30)
    plt.title(f'Гистограмма {col}')
    plt.subplot(1, 2, 2)
    df[col].plot(kind='box')
    plt.title(f'Boxplot {col}')
    plt.tight_layout()
    plt.show()

#ДОПОЛНИТЕЛЬНЫЕ ГРАФИКИ ПО SCORE
# 1) Распределение Score
df['Score'].plot(kind='hist', bins=30, color='pink', edgecolor='black')
plt.xlabel('Индекс счастья')
plt.ylabel('Количество стран')
plt.title('Распределение индекса счастья')
plt.show()

# 2) Топ-30 самых счастливых стран
region = df.groupby("Country or region", as_index=False)["Score"].mean()
region = region.sort_values("Score").reset_index(drop=True)
top30 = region.tail(30)

fig, ax = plt.subplots(figsize=(10, 20))
ax.barh(range(len(top30.index)), top30["Score"], align='edge', height=.5,
color='MediumAquamarine')
ax.set_yticks(range(len(top30.index)))
ax.set_yticklabels(top30["Country or region"])
plt.ylim(0, len(top30.index))
plt.xlabel('Индекс счастья')
plt.ylabel('Страна или регион')
plt.title('Топ-30 самых счастливых стран')
plt.show()

# 3) Топ-30 самых несчастливых стран
bottom30 = region.head(30)
fig, ax = plt.subplots(figsize=(10, 20))
ax.barh(range(len(bottom30.index)), bottom30["Score"], align='edge', height=.5, color='Peru')
ax.set_yticks(range(len(bottom30.index)))
ax.set_yticklabels(bottom30["Country or region"])
plt.ylim(0, len(bottom30.index))
plt.xlabel('Индекс счастья')
plt.ylabel('Страна или регион')
plt.title('Топ-30 самых несчастливых стран')
```

```
plt.show()

outlier_cols = ['GDP per capita', 'Social support', 'Healthy life expectancy',
            'Freedom to make life choices', 'Generosity', 'Perceptions of corruption']

# Проверка на наличие выбросов по методу IQR
def iqr_outliers_report(data, cols, k=1.5):
    for col in cols:
        q1 = data[col].quantile(0.25)
        q3 = data[col].quantile(0.75)
        iqr = q3 - q1
        lower = q1 - k * iqr
        upper = q3 + k * iqr
        outliers = data[(data[col] < lower) | (data[col] > upper)]
        print(f"{col}: {len(outliers)} выбросов из {len(data)} строк")

print("Выбросы в данных:")
iqr_outliers_report(df, outlier_cols)

#Винсоризация выбросов
def winsorize_series(s, k=1.5):
    q1, q3 = s.quantile([0.25, 0.75])
    iqr = q3 - q1
    lower, upper = q1 - k*iqr, q3 + k*iqr
    return s.clip(lower, upper)

for col in outlier_cols :
    if col in df.columns:
        df[col] = winsorize_series(df[col])

print("Выбросы после винзоризации:")
iqr_outliers_report(df, outlier_cols)

#Корреляционная матрица
new_df1 = df[['Score', 'GDP per capita', 'Social support', 'Healthy life expectancy',
            'Freedom to make life choices', 'Generosity', 'Perceptions of corruption']].copy()
plt.figure(figsize=(10, 8))
sns.heatmap(new_df1.corr(), annot=True, cbar=False)
plt.title("Корреляционная матрица")
plt.show()

# Построение и обучение моделей машинного обучения

#Признаки и целевая переменная
X_all = df.drop(columns=['Country or region', 'Score'])
y = df['Score']
```

```
x_train, x_test, y_train, y_test = train_test_split(X_all, y, random_state=2, test_size=0.2)

# Метрики
def check_metrics(y_pred, y_test, name):
    print('--', name, '--')
    print('R²:', r2_score(y_test, y_pred))
    print('MAPE:', mean_absolute_percentage_error(y_test, y_pred))
    print('MAE:', mean_absolute_error(y_test, y_pred))
    print('RMSE:', np.sqrt(mean_squared_error(y_test, y_pred)))

#Функции моделей
def model_linreg(x_train, x_test, y_train, y_test):
    scaler = StandardScaler()
    x_train_scaled = scaler.fit_transform(x_train)
    x_test_scaled = scaler.transform(x_test)

    reg = LinearRegression()
    reg.fit(x_train_scaled, y_train)
    y_pred = reg.predict(x_test_scaled)
    check_metrics(y_pred, y_test, "Линейная регрессия")

def model_ridge(x_train, x_test, y_train, y_test, alpha=1.0):
    scaler = StandardScaler()
    x_train_scaled = scaler.fit_transform(x_train)
    x_test_scaled = scaler.transform(x_test)

    reg = Ridge(alpha=alpha)
    reg.fit(x_train_scaled, y_train)
    y_pred = reg.predict(x_test_scaled)
    check_metrics(y_pred, y_test, "Ridge регрессия")

def model_poly(x_all, y, degree=2):
    poly = PolynomialFeatures(degree=degree, include_bias=False)
    X_poly = poly.fit_transform(x_all)
    x_train_p, x_test_p, y_train_p, y_test_p = train_test_split(X_poly, y, random_state=2,
test_size=0.2)

    scaler = StandardScaler()
    x_train_p_scaled = scaler.fit_transform(x_train_p)
    x_test_p_scaled = scaler.transform(x_test_p)

    reg = LinearRegression()
    reg.fit(x_train_p_scaled, y_train_p)
    y_pred = reg.predict(x_test_p_scaled)
    check_metrics(y_pred, y_test_p, f"Полиномиальная регрессия (deg={degree})")
```

```
def model_rf(x_train, x_test, y_train, y_test):
    print("\nОбучаем Random Forest с подбором гиперпараметров...")
    rf = RandomForestRegressor(random_state=2)   # сначала создаём модель
    param_grid_rf = {
        'n_estimators': [100, 200, 300],
        'max_depth': [None, 5, 10],
        'min_samples_split': [2, 5],
        'min_samples_leaf': [1, 2]
    }
    grid = GridSearchCV(rf, param_grid_rf, cv=5, scoring='r2', n_jobs=-1)
    grid.fit(x_train, y_train)
    print("Лучшие параметры RandomForest:", grid.best_params_)
    y_pred = grid.predict(x_test)
    check_metrics(y_pred, y_test, "Случайный лес")

def model_knn(x_train, x_test, y_train, y_test):
    print("\nОбучаем KNN с подбором гиперпараметров...")
    knn = KNeighborsRegressor()
    pipe = Pipeline([
        ("scaler", StandardScaler()),
        ("knn", knn)
    ])
    param_grid_knn = {
        'knn__n_neighbors': [3, 5, 7, 9, 11],
        'knn__weights': ['uniform', 'distance']
    }
    grid = GridSearchCV(pipe, param_grid_knn, cv=5, scoring='r2', n_jobs=-1)
    grid.fit(x_train, y_train)
    print("Лучшие параметры KNN:", grid.best_params_)
    y_pred = grid.predict(x_test)
    check_metrics(y_pred, y_test, "KNN (GridSearchCV)")

model_linreg(x_train, x_test, y_train, y_test)
model_ridge(x_train, x_test, y_train, y_test)
model_poly(X_all, y, degree=2)
model_rf(x_train, x_test, y_train, y_test)
model_knn(x_train, x_test, y_train, y_test)
```