

Влияние социально-экономических факторов на уровень убийств в городе Детройт (1961-1973): анализ и стратегии преступностей

Бородулин Андрей Вадимович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

Исследование направлено на анализ влияния социально-экономических факторов на уровень убийств в городе Детройт в период с 1961 по 1973 год. Акцент сделан на исследовании взаимосвязей между уровнем преступности и различными переменными, такими как количество полицейских, уровень безработицы и другие социальные показатели. Методы анализа включают разведочный анализ данных и построение нескольких моделей регрессии. Результаты подчеркивают важность определенных факторов в формировании образа преступности и предоставляют основу для разработки эффективных стратегий по снижению уровня преступности в городской среде.

Ключевые слова: Уровень убийств, Социально-экономические факторы, Детройт, Регрессионный анализ, Стратегии преступности, Социальная безопасность, Полицейские ресурсы, Безработица, Разведочный анализ данных, Моделирование преступности.

Influence of socio-economic factors on homicide rates in Detroit (1961-1973): analysis and crime prevention strategies

Borodulin Andrei Vadimovich

Sholom-Aleichem Priamursky State University

Student

Abstract

The research aims to analyze the impact of socio-economic factors on the homicide rate in the city of Detroit during the period from 1961 to 1973. The emphasis is placed on exploring the relationships between crime rates and various variables, such as the number of police officers, unemployment rate, and other social indicators. Analytical methods include exploratory data analysis and the construction of multiple regression models. The results underscore the importance of specific factors in shaping the crime landscape and provide a foundation for developing effective strategies to reduce crime levels in an urban environment.

Keywords: Homicide rate, Socio-economic factors, Detroit, Regression analysis, Crime prevention strategies, Social-security, Police resources, Unemployment, Exploratory data analysis, Crime modeling

1 Введение

1.1 Актуальность

В современном обществе преступность и уровень преступности представляют серьезную социальную проблему. Исследование факторов, влияющих на уровень убийств, позволяет выявить возможные тенденции и разработать меры по их предотвращению. В условиях современного города, такого как Детройт в период 1961-1973 гг., необходимо понять взаимосвязи между социально-экономическими факторами и уровнем преступности, чтобы разработать эффективные стратегии по улучшению общественной безопасности.

1.2 Обзор исследований

В своей работе Д.В. Кузнецов рассмотрел события, включая "лос-анджелесский бунт" 1992 года и беспорядки 1960-х годов, выявив их связь с расовыми противоречиями и подчеркнув уникальность и масштабность беспорядков в 2020 году, в которых широко участвовали различные социокультурные группы и которые оказали значительное воздействие на общественно-политическую обстановку в США [1]. А.А.Шумейко посвятила статью определив социально-значимые факторы, влияющие на число зарегистрированных преступлений, а также с помощью факторного анализа выявлены и исключены из построенных математических моделей некачественные и несущественные факторы. Результатом работы являются эконометрические модели, описывающие изменение уровня преступности в зависимости от выявленных качественных факторов, пригодные для дальнейшего исследования и прогнозирования преступности на основе социально-экономических показателей[2].А.З. Шевченко рассмотрел в своей публикации основные проблемы упадка экономики и социальной дезорганизации, и попытался разобраться в их появлении, проведя рассуждение над конкретными причинами, такими, как: упадок экономики, недостаток рабочих мест, снижение инвестиций, социальная дезорганизация, увеличение преступности, снижение уровня образования[3].В своей работе В.Е. Квашис рассмотрел основные тенденции, сравнил показатели преступности в США и России в период с 1991 по 2011 г, а также обсудил меры по противодействию преступности[4]. Э.Л. Нитобург анализирует эволюцию расовой ситуации в США с периода движения за гражданские права до начала XXI века, рассматривая влияние законов о гражданских правах и проблем расовой дискриминации в различных областях общества, что может быть связано с вашим исследованием на тему преступности и факторов, влияющих на нее [5].

1.3 Цель исследования

Исследование направлено на анализ влияния социально-экономических факторов на уровень убийств в городе Детройт в период с 1961 по 1973 год. Акцент сделан на исследовании взаимосвязей между уровнем преступности и

различными переменными, такими как количество полицейских, уровень безработицы, социальные и экономические показатели.

2 Материалы и методы

2.1 Исходные данные

Исходные данные для исследования представляют собой датасет, содержащий информацию о преступности в городе Детройт в период с 1961 по 1973 год. Данные включают различные социально-экономические показатели, такие как количество полицейских, уровень безработицы, количество рабочих в производстве, лицензии на оружие и другие. Ссылка на датасет: <http://lib.stat.cmu.edu/datasets/detroit>.

2.2 Методы анализа данных

Исследование начинается с разведочного анализа данных (EDA), включающего ознакомление с основными статистиками, визуализацию распределения целевой переменной (убийства) и построение корреляционной матрицы для выявления взаимосвязей между переменными.

Методы анализа включают разведочный анализ данных, построение нескольких моделей регрессии, а также оценку и сравнение их производительности. Результаты исследования подчеркивают важность определенных факторов в формировании образа преступности и предоставляют основу для разработки эффективных стратегий по снижению уровня преступности в городской среде.

Исследования осуществлялось с использованием языка программирования Python и ряда библиотек для анализа данных и машинного обучения. В частности, для загрузки, обработки и визуализации данных были задействованы библиотеки Pandas, Matplotlib и Seaborn. Для построения и оценки моделей машинного обучения были использованы библиотеки Scikit-learn и XGBoost. Применение этих технологий позволило провести всесторонний анализ данных, построить несколько моделей регрессии и оценить их производительность.

2.3 Выбор моделей

Для построения и оценки моделей использованы следующие методы:

Линейная регрессия: Простая модель, позволяющая оценить линейные отношения между признаками и целевой переменной.

Дерево решений: Нелинейная модель, способная улавливать сложные зависимости в данных.

Случайный лес: Ансамблевая модель, объединяющая несколько деревьев решений для повышения обобщающей способности.

Градиентный бустинг: Ансамблевая модель, последовательно обучающая слабые модели и уделяющая внимание ошибкам предыдущих моделей.

XGBoost: Эффективный алгоритм градиентного бустинга, придающий больший вес ошибкам предыдущих моделей.

2.4 Оценка моделей

Оценка моделей производится на основе метрик качества, таких как среднеквадратичная ошибка (RMSE) и коэффициент детерминации (R2 Score). Модели сравниваются между собой, а также проводится оценка значимости признаков с использованием модели случайного леса.

2.5 Подготовка данных

Для обеспечения стабильности и сравнимости результатов, данные проходят процесс нормализации перед обучением моделей. Фичи масштабируются с использованием стандартного скалирования.

2.6 Разделение данных

Данные разделяются на обучающий и тестовый наборы с использованием метода `train_test_split` для эффективного оценивания производительности моделей на новых данных.

2.7 Использование библиотек и инструментов

Для реализации методов анализа данных и построения моделей использованы библиотеки `pandas`, `matplotlib`, `seaborn`, `scikit-learn` и `XGBoost` в языке программирования Python.

3 Результаты и обсуждения

Весь анализ будет выполнен в среде Google Colab. Для начала необходимо импортировать необходимые библиотеки и выполнить считывание csv-файла с исходными данными:

```
# Импортирование библиотек
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Загрузка данных из CSV файла
df = pd.read_csv("homicide_data.csv")
```

3.1 Подготовка данных и исследовательский анализ

Исследование началось с первичного анализа данных. Были выведены первые строки данных, общая информация и статистики описательной статистики:

```
# Вывод первых 5-ти строк DataFrame
print(df.head())

# Общая информация о данных
print(df.info())

# Статистики описательной статистики
print(df.describe())
```

FTP	UEMP	MAN	LIC	GR	CLEAR	WM	NMAN	GOV	HE	\
-----	------	-----	-----	----	-------	----	------	-----	----	---

0	260.35	11.0	455.5	178.15	215.98	93.4	558724.0	538.1	133.9
2.98									
1	269.80	7.0	480.2	156.41	180.48	88.5	538584.0	547.6	137.6
3.09									
2	272.04	5.2	506.1	198.02	209.57	94.4	519171.0	562.8	143.6
3.23									
3	272.96	4.3	535.8	222.10	231.67	92.0	500457.0	591.0	150.3
3.33									
4	272.51	3.5	576.0	301.92	297.65	91.0	482418.0	626.1	164.3
3.46									

```
      WE      HOM      ACC      ASR
0  117.18    8.60   39.17  306.18
1  134.02    8.90   40.27  315.16
2  141.68    8.52   45.31  277.53
3  147.98    8.89   49.51  234.07
4  159.85   13.07   55.05  230.84
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13 entries, 0 to 12
Data columns (total 14 columns):
#   Column  Non-Null Count  Dtype
---  -
0    FTP      13 non-null    float64
1   UEMP      13 non-null    float64
2    MAN      13 non-null    float64
3    LIC      13 non-null    float64
4    GR       13 non-null    float64
5  CLEAR     13 non-null    float64
6    WM       13 non-null    float64
7   NMAN     13 non-null    float64
8    GOV     13 non-null    float64
9    HE      13 non-null    float64
10   WE       13 non-null    float64
11   HOM      13 non-null    float64
12   ACC      13 non-null    float64
13   ASR      13 non-null    float64
dtypes: float64(14)
memory usage: 1.5 KB
None
```

	FTP	UEMP	MAN	LIC	GR
CLEAR \					
count	13.000000	13.000000	13.000000	13.000000	13.000000
13.000000					
mean	304.511538	5.792308	556.446154	537.506923	545.659231
81.446154					
std	46.811656	2.359188	49.822161	316.415140	311.031612
12.659227					
min	260.350000	3.200000	455.500000	156.410000	180.480000
58.900000					
25%	269.800000	3.900000	535.800000	222.100000	231.670000
73.900000					
50%	272.960000	5.200000	569.300000	583.170000	616.540000
87.400000					
75%	341.430000	7.100000	596.900000	794.900000	750.430000
91.000000					
max	390.190000	11.000000	613.500000	1131.210000	1029.750000
94.400000					

	WM	NMAN	GOV	HE	WE \
count	13.000000	13.000000	13.000000	13.000000	13.000000
mean	452507.538462	673.923077	185.769231	3.947692	169.970769

std	64568.123896	94.773398	37.036185	0.966559	42.511238
min	359647.000000	538.100000	133.900000	2.910000	117.180000
25%	401518.000000	591.000000	150.300000	3.230000	141.680000
50%	448267.000000	686.200000	187.500000	3.600000	157.190000
75%	500457.000000	755.300000	223.800000	4.470000	178.740000
max	558724.000000	819.800000	230.900000	5.760000	258.050000

	HOM	ACC	ASR
count	13.000000	13.000000	13.000000
mean	25.126923	46.923077	311.950000
std	16.385435	5.139600	73.091153
min	8.520000	39.170000	217.990000
25%	8.900000	44.170000	277.530000
50%	21.360000	45.800000	306.180000
75%	37.390000	50.620000	323.030000
max	52.330000	55.050000	473.010000

Затем была построена корреляционная матрица (рис. 1) для оценки взаимосвязей между признаками:

```
# Корреляционная матрица
correlation_matrix=df.corr()
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm")
plt.title("Корреляционная матрица")
plt.show()
```

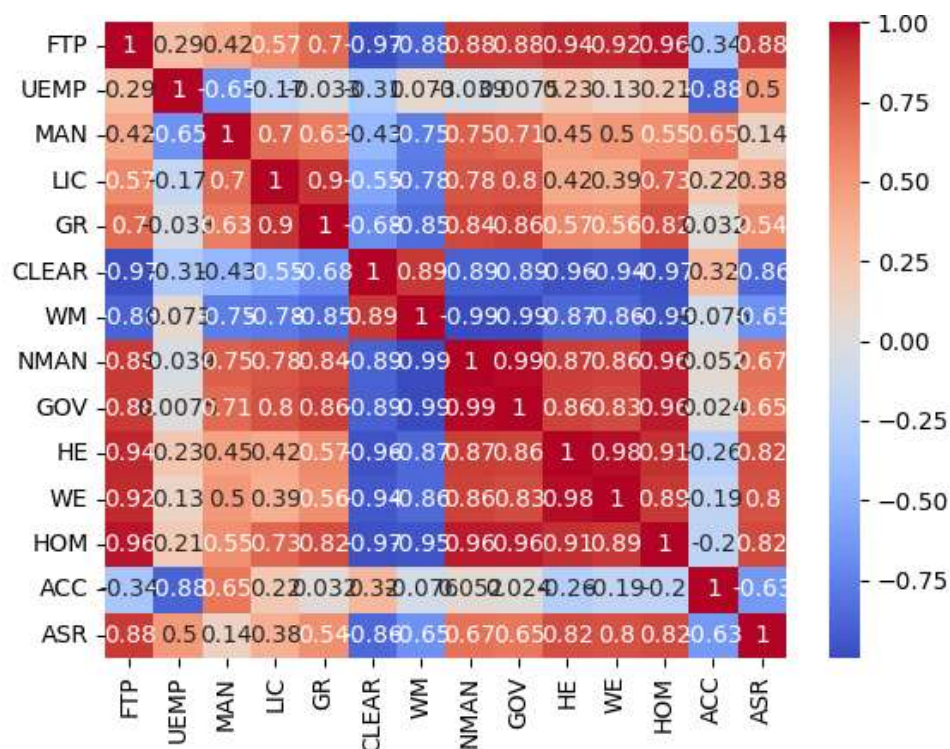


Рис. 1 - Корреляционная матрица

Корреляционная матрица, представленная на графике, помогает выявить взаимосвязи между различными переменными в исследуемом датасете. Заметные значения корреляции могут указывать на потенциальные зависимости между парами признаков. Несколько высоких значений корреляции:

MAN (Количество рабочих в производстве) и NMAN (Количество работников, не связанных с производством):

Высокая корреляция между этими двумя признаками может свидетельствовать о том, что уровень рабочей силы в промышленности и несвязанных с производством областях схож.

"HE" (средняя почасовая заработная плата) и "WE" (средняя недельная заработная плата) также имеют высокую положительную корреляцию, что логично, учитывая, что недельная заработная плата зависит от почасовой.

CLEAR (% раскрытых убийств) и ACC (Смертность в результате несчастных случаев):

Наличие связи между процентом раскрытых убийств и смертностью от несчастных случаев может быть интересным явлением и требовать дополнительного анализа.

В связи с этими наблюдениями рекомендуется рассмотреть возможность удаления одного из признаков из каждой из этих пар с целью избежания мультиколлинеарности в моделях регрессии.

Удалим один из признаков из пары "MAN" (количество рабочих в производстве) и "NMAN" (количество работников, не связанных с производством). Также удалим признак WE. В данном случае, "HE" и "WE" представляют схожую информацию, поэтому можно выбрать один из них для удаления.

```
df = df.drop("NMAN", axis=1)
df = df.drop("WE", axis=1)
```

Также, было изучено распределение целевой переменной "НОМ"(Количество убийств на 100 000 населения) (рис. 2):

```
# Распределение целевой переменной "НОМ"
sns.histplot(df["НОМ"], kde=True)
plt.title("Распределение убийств")
plt.xlabel("Число убийств на 100 000 населения")
plt.show()
```

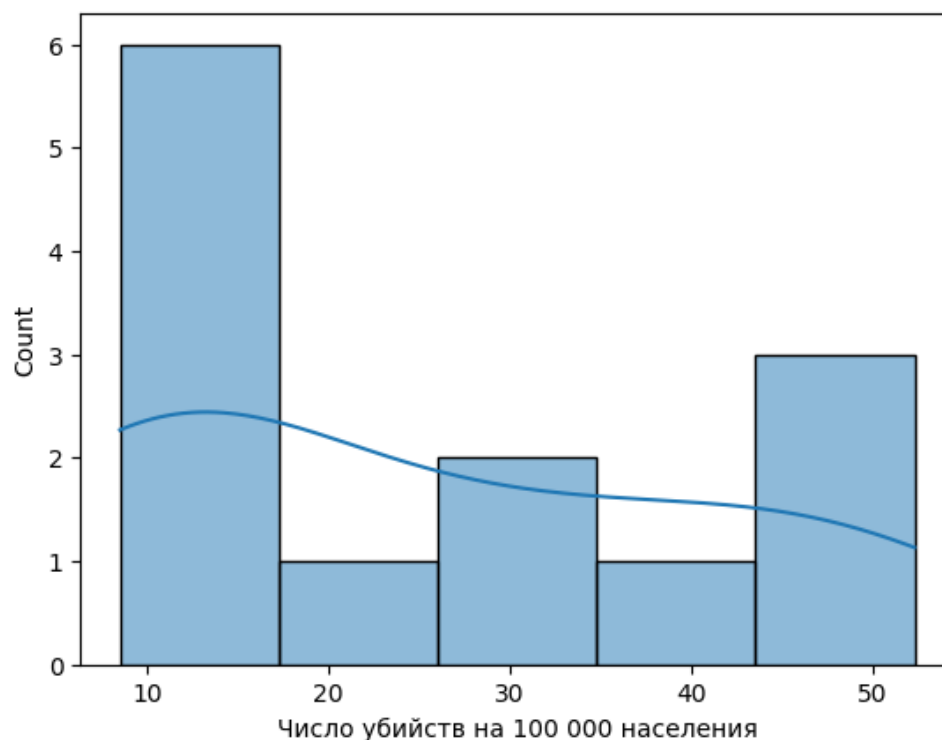


Рис. 2–График распределения убийств на 100 000 населения

График демонстрирует частоту различных значений убийств, а KDE добавляет сглаженную линию, представляющую оценку плотности вероятности для переменной.

Распределение смещено вправо: Большинство районов имеют низкий уровень убийств, но есть несколько районов с более высокими значениями. Это может свидетельствовать о том, что уровень убийств в городе неоднороден, и существуют районы с более высокими уровнями преступности. График имеет тяжелый хвост вправо, что может указывать на наличие отдельных районов или событий, где уровень убийств значительно выше среднего. Возможно, есть некоторые выбросы с очень высокими значениями убийств, что может быть важным для анализа и принятия мер для улучшения безопасности в этих районах.

Были построены scatterplot для каждой пары фичей:

```
features = df.drop('HOM', axis=1)

# Построение scatterplot для каждой пары фичей
sns.pairplot(features)
plt.show()
```

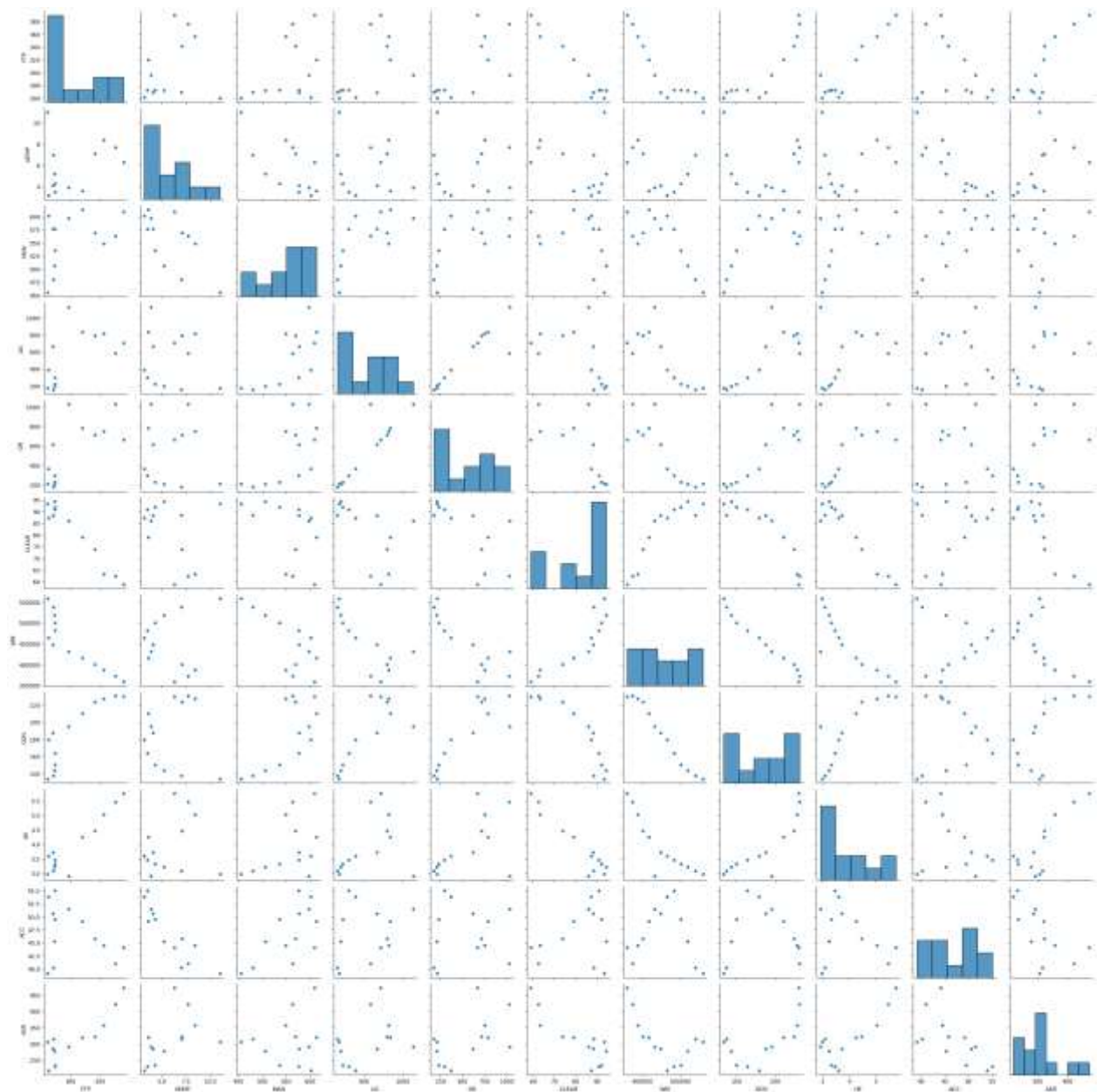
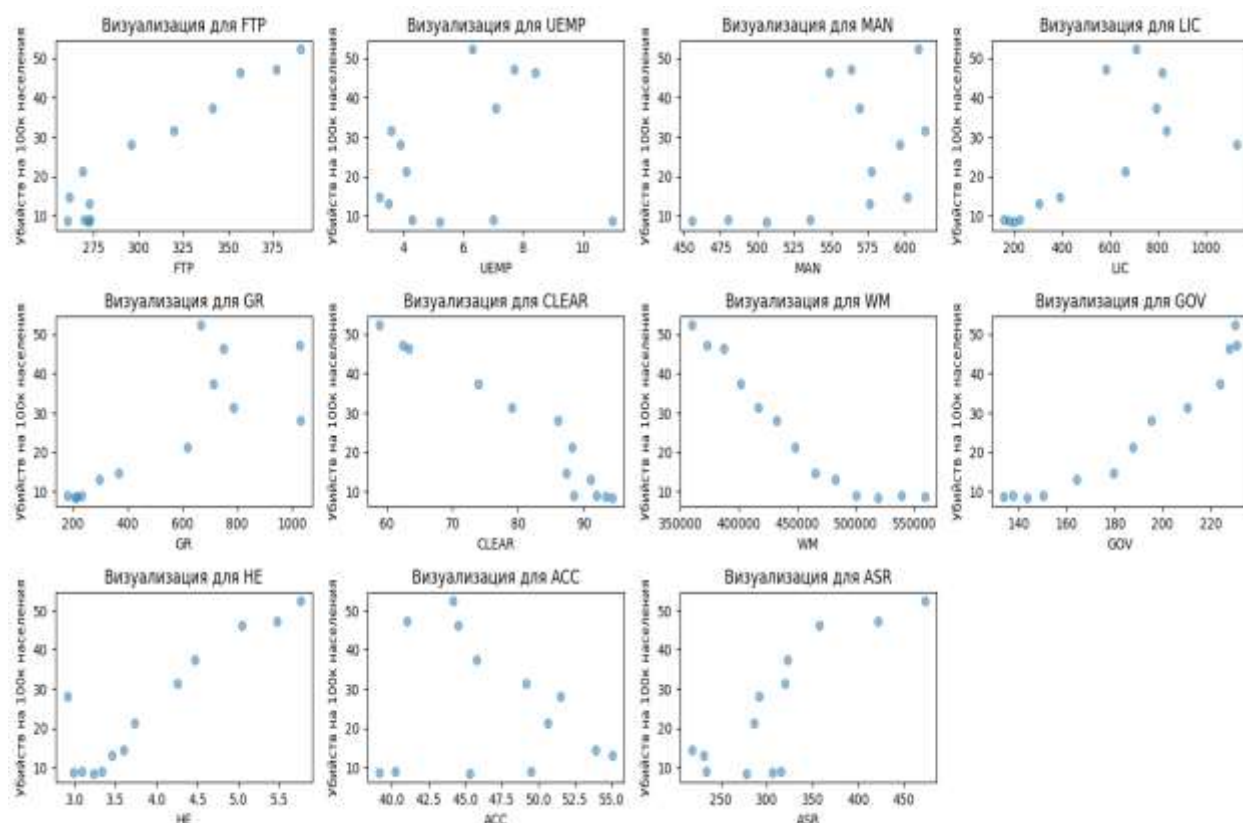


Рис. 3–Метод визуализации Scatterplot для каждой фичи попарно

Помимо этого, реализована визуализация для каждой фичи по отдельности:

```
plt.figure(figsize=(15,10))
for i, feature in enumerate(features.columns):
    plt.subplot(4,4, i+1)
    plt.scatter(features[feature], y, alpha=0.5)
    plt.title(f"Визуализация для {feature}")
    plt.xlabel(feature)
    plt.ylabel("Убийств на 100к населения")
    plt.tight_layout()
    plt.show()
```



Эти визуализации помогают более детально изучить взаимосвязи и распределения между признаками и целевой переменной.

3.2 Разделение данных и обучение моделей

Далее выделим признаки (X) и целевую переменную (y), следом за этим данные будут разделены на обучающий и тестовый наборы, а затем использованы для обучения нескольких моделей регрессии:

Выделение признаков (X) и целевой переменной (y)

```
X = df.drop("HOM", axis=1)
```

```
y = df["HOM"]
```

```
# Вывод первых 5 строк DataFrame для X и y
```

```
print ("X:")
```

```
print(X.head())
```

```
print ("\ny: ")
```

```
print(y.head())
```

X:

FTP	UEMP	MAN	LIC	GR	CLEAR	WM	GOV	HE	ACC	\
0260.	3511.	0455.	5178.	15215.	9893.	4558724.	0133.	92.	9839.	17
1269.	807.	0480.	2156.	41180.	4888.	5538584.	0137.	63.	0940.	27
2272.	045.	2506.	1198.	02209.	5794.	4519171.	0143.	63.	2345.	31
3272.	964.	3535.	8222.	10231.	6792.	0500457.	0150.	33.	3349.	51
4272.	513.	5576.	0301.	92297.	6591.	0482418.	0164.	33.	4655.	05
ASR										
0306.	18									
1315.	16									

2277.53
3234.07
4230.84

y:
08.60
18.90
28.52
38.89
413.07
Name: HOM, dtype: float64

```
# Разделение на обучающий и тестовый наборы
X_train, X_test, y_train, y_test = train_test_split(X,
y, test_size=0.2, random_state=42)

# Инициализация моделей
linear_model = LinearRegression()
tree_model = DecisionTreeRegressor(random_state=42)
forest_model = RandomForestRegressor(random_state=42)
gradient_boosting_model = GradientBoostingRegressor(random_state=42)
normalized_linear_model = LinearRegression()
xgboost_model = XGBRegressor(random_state=42)

# Обучение моделей на стандартных данных
linear_model.fit(X_train, y_train)
tree_model.fit(X_train, y_train)
forest_model.fit(X_train, y_train)
gradient_boosting_model.fit(X_train, y_train)
xgboost_model.fit(X_train, y_train)
```

3.3. Оценка моделей и визуализация результатов

После обучения моделей были получены предсказания и оценены их результаты:

```
# Оценка моделей
linear_rmse = mean_squared_error(y_test, linear_predictions, squared=False)
tree_rmse = mean_squared_error(y_test, tree_predictions, squared=False)
forest_rmse = mean_squared_error(y_test, forest_predictions, squared=False)
gradient_boosting_rmse = mean_squared_error(y_test, gradient_boosting_predictions, squared=False)
xgboost_rmse = mean_squared_error(y_test, xgboost_predictions, squared=False)

linear_r2 = r2_score(y_test, linear_predictions)
tree_r2 = r2_score(y_test, tree_predictions)
forest_r2 = r2_score(y_test, forest_predictions)
gradient_boosting_r2 = r2_score(y_test, gradient_boosting_predictions)
xgboost_r2 = r2_score(y_test, xgboost_predictions)

# Вывод результатов для стандартных данных
print("Результаты на стандартных данных:")
print(f"Линейная регрессия RMSE: {linear_rmse}, R2 Score: {linear_r2}")
print(f"Дерево решений RMSE: {tree_rmse}, R2 Score: {tree_r2}")
print(f"Случайный лес RMSE: {forest_rmse}, R2 Score: {forest_r2}")
print(f"Градиентный бустинг RMSE: {gradient_boosting_rmse}, R2 Score: {gradient_boosting_r2}")
print(f"XGBoost RMSE: {xgboost_rmse}, R2 Score: {xgboost_r2}")
```

В результате оценки были получены значения метрик для каждой модели. Особо стоит отметить, что модели случайного леса и XGBoost показали наилучшие результаты, что видно по наименьшему значению RMSE и наивысшему R2 Score:

Линейная регрессия RMSE: 14.556948294427121, R2 Score: 0.2115779271815348
 Дерево решений RMSE: 15.889098988509911, R2 Score: 0.06067336542402413
 Случайный лес RMSE: 3.2656734813307016, R2 Score: 0.9603207353794617
 Градиентный бустинг RMSE: 4.534611818517812, R2 Score: 0.9234935013574261
 XGBoost RMSE: 3.6055138584217996, R2 Score: 0.9516326343582584

3.4. Визуализация значимости признаков

Для оценки значимости признаков в модели случайного леса была построена визуализация (рис. 5):

```
# Оценка значимости признаков в случайном лесе
feature_importances=forest_model.feature_importances_

# Создание DataFrame для удобного отображения
feature_importance_df=pd.DataFrame({"Признак": X.columns, "Важность":
feature_importances})
feature_importance_df=feature_importance_df.sort_values(by="Важность",
ascending=False)

# Визуализация значимости признаков
plt.figure(figsize=(10,6))
sns.barplot(x="Важность", y="Признак", data=feature_importance_df,
palette="viridis")
plt.title("Важность признака в \"Случайный лес\"")
plt.show()
```

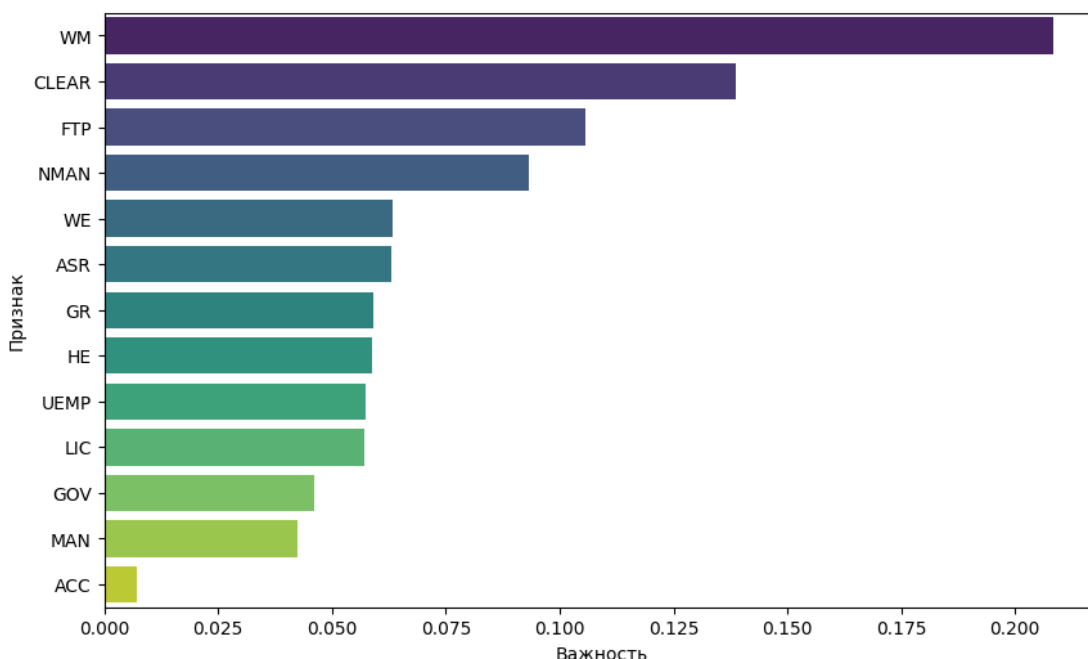


Рис. 5—Важность признака в «Случайный лес»

На графике важности признаков, построенном на основе модели случайного леса, видно, что наибольший вклад в предсказание уровня убийств

вносит несколько ключевых факторов. Среди них можно выделить "CLEAR" (процент раскрытых убийств), "MAN" (количество рабочих в производстве), "GR" (количество регистраций на оружие на 100,000 населения) и "ACC" (смертность в результате несчастных случаев на 100,000 населения).

Эти результаты могут указывать на важность эффективности полицейских мероприятий (CLEAR), общего числа рабочих в производстве (MAN), регулирования доступа к оружию (GR) и уровня безопасности в среде, что включает в себя смертность от несчастных случаев (ACC). Понимание влияния этих факторов может быть ключевым при разработке стратегий по снижению уровня преступности в городской среде.

3.5. Нормализация данных

Для дальнейшего улучшения производительности моделей была проведена нормализация данных. Нормализация в данном контексте включает в себя приведение значений признаков к стандартным диапазонам, что особенно важно при использовании моделей, чувствительных к масштабу данных, таких как линейная регрессия. Ниже представлен код, осуществляющий нормализацию данных и последующее обучение моделей на нормализованных данных:

```
# Нормализацияфичей
scaler =StandardScaler()
X_train_scaled=scaler.fit_transform(X_train)
X_test_scaled=scaler.transform(X_test)

# Обучениemodelейнанормализованныхданных
normalized_linear_model.fit(X_train_scaled,y_train)
tree_model.fit(X_train_scaled,y_train)
forest_model.fit(X_train_scaled,y_train)
gradient_boosting_model.fit(X_train_scaled,y_train)
xgboost_model.fit(X_train_scaled,y_train)
```

3.6. Сравнение результатов на стандартных и нормализованных данных

После обучения моделей на нормализованных данных, произведено сравнение их результатов с производительностью моделей на стандартных данных. Для наглядности результаты обеих групп моделей были выведены и проанализированы.

```
# Оценкамodelейнанормализованныхданных
normalized_linear_rmse=mean_squared_error(y_test,normalized_linear_predictions, squared=False)
tree_rmse=mean_squared_error(y_test,tree_predictions, squared=False)
forest_rmse=mean_squared_error(y_test,forest_predictions, squared=False)
gradient_boosting_rmse=mean_squared_error(y_test,gradient_boosting_predictions, squared=False)
xgboost_rmse=mean_squared_error(y_test,xgboost_predictions, squared=False)

r2_score_normalized_linear =
r2_score(y_test,normalized_linear_predictions)
r2_score_tree = r2_score(y_test,tree_predictions)
r2_score_forest = r2_score(y_test,forest_predictions)
```

```
r2_score_gradient_boosting =  
r2_score(y_test, gradient_boosting_predictions)  
r2_score_xgboost = r2_score(y_test, xgboost_predictions)  
  
# Вывод результатов  
print(f"На нормализованных данных:")  
print(f"Линейная регрессия RMSE: {normalized_linear_rmse}, R2 Score:  
{r2_score_normalized_linear}")  
print(f"Дерево решений RMSE: {tree_rmse}, R2 Score: {r2_score_tree}")  
print(f"Случайный лес RMSE: {forest_rmse}, R2 Score: {r2_score_forest}")  
print(f"Градиентный бустинг RMSE: {gradient_boosting_rmse}, R2 Score:  
{r2_score_gradient_boosting}")  
print(f"XGBoost RMSE: {xgboost_rmse}, R2 Score: {r2_score_xgboost}")
```

Сравнение результатов на стандартных и нормализованных данных позволяет оценить влияние нормализации на производительность моделей и выделить наилучшие методы для дальнейшего анализа и использования.

Оценка работы моделей на нормализованных данных показала такие результаты:

```
Линейная регрессия RMSE: 4.2919488525577805, R2 Score: 0.931462671658414  
Дерево решений RMSE: 15.889098988509911, R2 Score: 0.06067336542402413  
Случайный лес RMSE: 3.2656734813307016, R2 Score: 0.9603207353794617  
Градиентный бустинг RMSE: 4.534611818517812, R2 Score: 0.9234935013574261  
XGBoost RMSE: 3.6055138584217996, R2 Score: 0.9516326343582584
```

После проведения нормализации данных результаты на линейной регрессии существенно улучшились: RMSE уменьшилось с 14.56 до 4.29, а R2 Score увеличился с 0.21 до 0.93. Это говорит о том, что нормализация положительно влияет на производительность линейной регрессии, улучшая точность предсказаний.

В то время как для моделей дерева решений, случайного леса, градиентного бустинга и XGBoost результаты остались практически неизменными после нормализации, что может свидетельствовать о том, что эти модели менее чувствительны к различиям в масштабе признаков.

В целом, результаты говорят в пользу того, что для линейных моделей, особенно линейной регрессии, нормализация данных может значительно улучшить качество предсказаний, в то время как для некоторых других моделей этот эффект может быть менее выраженным.

Выводы

Исследование направлено на анализ влияния социально-экономических факторов на уровень убийств в городе Детройт в период с 1961 по 1973 год. Акцент сделан на изучении взаимосвязей между уровнем преступности и различными переменными, такими как количество полицейских, уровень безработицы и другие социальные показатели.

Методы анализа включают разведочный анализ данных и построение нескольких моделей регрессии. Результаты подчеркивают важность определенных факторов в формировании образа преступности и

предоставляют основу для разработки эффективных стратегий по снижению уровня преступности в городской среде.

Проведенные исследования с применением различных моделей машинного обучения, таких как линейная регрессия, дерево решений, случайный лес, градиентный бустинг и XGBoost, позволили выявить, что случайный лес и градиентный бустинг демонстрируют наилучшие результаты в предсказании уровня убийств. Это подтверждается наименьшей среднеквадратичной ошибкой (RMSE) и наивысшим коэффициентом детерминации (R2 Score) по сравнению с другими моделями.

Важность признаков, определенная в модели случайного леса, указывает на то, что факторы, такие как количество полицейских на 100 000 населения и процент раскрытых убийств, оказывают значительное влияние на уровень преступности.

Исследование предоставляет базу для дальнейших стратегических мер по управлению уровнем преступности в городской среде, учитывая социально-экономические факторы, выявленные в данном анализе.

Библиографический список

1. Кузнецов Д.В. На краю пропасти?: к характеристике общественно-политической ситуации в США в 2020 году// Американистика на дальнем востоке. 2020. С. 56.
2. Шумейко А.А. Факторный анализ преступности на основе социально-экономических показателей // Проблемы современной экономики: материалы VI МНК (г. Казань, август 2017 г.), 2017. С. 22-26.
3. Шевченко А.З. Проблемы Детройта упадок экономики и социальная дезорганизация URL: <https://spawnalley.ru/problems-detroit-a-upadok-ekonomiki-i-sotsialnaya-dezorganizatsiya/> (дата обращения: 22.11.2023).
4. Квашиш В.Е. Преступность в США: тенденции, причины, меры противодействия. // Научный портал МВД России. 2013. С. 27-38
5. Нитобург Э.Л. О некоторых аспектах расовой проблемы в США на рубеже XX-XXI веков // Новая и новейшая история. 2008. С. 69-86
6. Vayani A. Exploratory Data Analysis – Police Killings. URL: <https://www.kaggle.com/code/ashmalvayani/exploratory-data-analysis-police-killings> (дата обращения: 22.11.2023).
7. Ella S. Пример анализа данных с использованием моделей KNN, SVM и SVM с гиперпараметрами. URL: <https://www.e-du.ru/2021/10/data-science-sample-knn-svm.html> (дата обращения: 22.11.2023).
8. К ВВ. Разведочный анализ данных в Python. URL: <https://waksoft.susu.ru/2021/07/23/issledovatel'skij-analiz-dannyh-v-python-rukovodstvo-dlya-novichkov-na-2021-god/> (дата обращения: 22.11.2023).