1830

Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский государственный технический университет имени Н.Э. Баумана

(национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)

ЦЕНТР ПРО	<u>ГОЛОВНОЙ</u> ФЕССИОНАЛЬНОЙ ЯМИ ЗДОРОВЬЯ					И МЕТОДИЧЕСКИЙ ОГРАНИЧЕННЫМИ
КАФЕДРА	СИСТЕМЫ ОБРАБС	ТКИ ИНФОРМ	<u>МАЦИИ И</u>	УПРАВЛ	ЕНИ.	<u>R</u>
РАСЧ	ЕТНО-ПО	ЯСНИТ	ЕЛЬ	КАН	3 <i>A</i>	ЛИСКА
К НА	УЧНО-ИСС	ЛЕДОВ.	АТЕЛ	ЬСКО	ΟЙ	РАБОТЕ
по дисциплине _		ологии маш	инного	обучени	<u>R</u> 1	
по теме	Решение зад	цачи машин	ного обу	учения		
	<u>5Ц-84Б</u> Группа)		(Подпись,	дата)	-	Е. С. Клеша (И.О.Фамилия)
Руководитель						Ю. Е. Гапанюк

(Подпись, дата)

(И.О.Фамилия)

Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ		
Заведующий кафедр	ой	<u>ИУ5</u>
		(Индекс)
	В. И.	Терехов
	(И.О.	Фамилия)
// \		2024 E

ЗАДАНИЕ

на выполнение научно-	-исследовательско	ой работы
по теме Решение задачи маш	инного обучения	
Студент группы <u>ИУ5Ц-84Б</u>		
<u>Клеша Егор Сергес</u> (Фамилия	евич	
Направленность НИР (учебная, исследовательс	ская, практическая, произв	одственная, др.)
исследовательская		
Источник тематики (кафедра, предприятие, Н	ИР) <u>кафедра</u>	
График выполнения НИР: 25% к <u>3</u> нед., 50	0% к <u>9</u> нед., 75% к <u>12</u> н	нед., 100% к <u>15</u> нед.
Техническое задание		
Решение задачи машинного обучения. Ре-	зультатом проекта являє	стся отчет, содержащий
описания моделей, тексты программ и рез	ультаты экспериментов.	
Оформление научно-исследовательской рабо		
Расчетно-пояснительная записка на <u>19</u> лист Перечень графического (иллюстративного) ма		, слайды и т.п.)
Дата выдачи задания «» 2024 г.		
Руководитель НИР	(Подпись, дата)	Ю. Е. Гапанюк (И.О.Фамилия)
Студент	(Подпись, дата)	Е. С. Клеша (И.О.Фамилия)

<u>Примечание</u>: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

СОДЕРЖАНИЕ

Вве,	дение	4
Осн	овная часть	6
1.	Импорт библиотек	6
2.	Загрузка данных.	6
	Изучение данных с целью обнаружения закономерностей. Создание рических представлений для более глубокого понимания информации. еделение и заполнение пропущенных значений в наборе данных	7
4.	Обработка данных	9
5. отно	Анализ корреляции данных с последующим извлечением выводов осительно возможности построения моделей машинного обучения 1	1
6. испо	Определение критериев оценки качества моделей для последующего ользования	2
7. или	Определение наилучших моделей для решения задачи классификации регрессии	
8. дата	Создание обучающего и тестового наборов данных из исходного исета	3
	Создание начального решения (baseline) для выбранных моделей без гройки гиперпараметров. Модели обучаются на обучающем наборе ных, а затем оцениваются по качеству на тестовом наборе	4
	Настройка гиперпараметров выбранных моделей с применением кроссидации	
	Применение оптимальных значений гиперпараметров для выбранных елей и оценка их качества сравнительно с базовыми моделями	6
12.	Вывод	7
Закл	ıючение1	8
Спи	сок использованных источников информации	9

Введение

Этот проект представляет собой самый всесторонний и полный источник информации о преступлениях, особенно убийствах, среди доступных на текущий момент в Соединенных Штатах. Он включает данные из Дополнительного отчета ФБР об убийствах с 1976 года по настоящее время, а также информацию, полученную в результате запросов по Закону о свободе информации о более чем 22 000 убийствах, которые не были включены в отчеты Министерства юстиции. Этот набор данных содержит информацию о возрасте, расе, поле, этнической принадлежности как жертв, так и преступников, а также об отношениях между ними и используемом оружии.

В качестве набора данных мы будем использовать набор данных, содержащий историю преступлений в США - https://www.kaggle.com/datasets/murderaccountability/homicide-reports.

Датасет состоит из одного файла homicideReports.csv Файл содержит следующие колонки:

- Record ID Идентификатор записи;
- Agency Code Код агентства;
- Agency Туре Тип агентства;
- City Город;
- State Государство;
- Year Год;
- Month Месяц;
- Incident Инцидент;
- Crime Туре Тип преступления;
- Crime Solved Раскрытое преступление;
- Victim Sex Пол жертвы;
- Victim Age Возраст жертвы;
- Victim Race Гонка жертв;
- Victim Ethnicity Этническая принадлежность жертвы;
- Perpetrator Sex Пол преступника;

- Perpetrator Age Возраст преступника;
- Perpetrator Race Paca преступников;
- Perpetrator Ethnicity Этническая принадлежность преступника;
- Relationship Взаимоотношения;
- Weapon Оружие;
- Victim Count Количество жертв;
- Perperator Count Количество преступников;
- Record Source Источник записи.

В данной работе будем решать задачи регрессии.

Основная часть

Ссылка на датасет: https://www.kaggle.com/datasets/yassine9/dataset-spine.

1. Импорт библиотек.

```
1 import numpy as np
    import pandas as pd
 3 import seaborn as sns
 4 import matplotlib.pyplot as plt
 5 import datetime
 6 from sklearn.preprocessing import MinMaxScaler
 7 from sklearn.linear_model import LinearRegression, LogisticRegression
 8 from sklearn.preprocessing import LabelEncoder
 9 from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsRegressor, KNeighborsClassifier
from sklearn.metrics import accuracy_score, balanced_accuracy_score
from sklearn.metrics import precision_score, recall_score, f1_score, classification_report
13 from sklearn.metrics import confusion_matrix
14 from sklearn.metrics import plot_confusion_matrix
15 from sklearn.model_selection import GridSearchCV
from sklearn.metrics import mean_absolute_error, mean_squared_error, mean_squared_log_error, median_absolute_err
from sklearn.metrics import roc_curve, roc_auc_score
from sklearn.svm import SVC, NuSVC, LinearSVC, OneClassSVM, SVR, NuSVR, LinearSVR
from sklearn.tree import DecisionTreeClassifier, DecisionTreeRegressor, export_graphviz
from sklearn.ensemble import RandomForestClassifier, RandomForestRegressor
from sklearn.ensemble import ExtraTreesClassifier, ExtraTreesRegressor
from sklearn.ensemble import GradientBoostingClassifier, GradientBoostingRegressor matplotlib inline
24 sns.set(style="ticks")
```

2. Загрузка данных.

Загрузим файлы датасета с помощью библиотеки Pandas.

Этот файл содержит данные в формате CSV, где информация разделена определенным символом, таким как запятая ",", точка с запятой ";" или табуляция. Перед чтением файла методом read_csv рекомендуется определить этот символ-разделитель, чтобы правильно указать его с помощью параметра sep. Это можно сделать, открыв файл в любом текстовом редакторе и просмотрев его структуру.

```
1 data = pd.read_csv('../datasets/homicideReports.csv', sep=',', low_memory=False)
```

Так как объем данных в наборе велик, мы решили ограничиться только первыми 15000 строками для упрощения анализа и обработки.

```
1 data = data.head(15000)
```

Из-за уменьшения количества строк, столбец Year перестает быть релевантным. В этом наборе данных будут использоваться только данные за один временной период.

3. Изучение данных с целью обнаружения закономерностей. Создание графических представлений для более глубокого понимания информации. Определение и заполнение пропущенных значений в наборе данных.

Основные характеристики набора данных.

Первые и последние 5 строк датасета.

1 da	ata													
	Record ID	Agency Code	Agency Name	Agency Type	City	State	Month	Incident	Crime Type	Crime Solved	 Victim Ethnicity	Perpetrator Sex	Perpetrator Age	Perpetr F
0	1	AK00101	Anchorage	Municipal Police	Anchorage	Alaska	January	1	Murder or Manslaughter	Yes	 Unknown	Male	15	N American/Al N
1	2	AK00101	Anchorage	Municipal Police	Anchorage	Alaska	March	1	Murder or Manslaughter	Yes	 Unknown	Male	42	V
2	3	AK00101	Anchorage	Municipal Police	Anchorage	Alaska	March	2	Murder or Manslaughter	No	 Unknown	Unknown	0	Unkn
3	4	AK00101	Anchorage	Municipal Police	Anchorage	Alaska	April	1	Murder or Manslaughter	Yes	 Unknown	Male	42	v
4	5	AK00101	Anchorage	Municipal Police	Anchorage	Alaska	April	2	Murder or Manslaughter	No	 Unknown	Unknown	0	Unkn
14995	14996	NY03030	New York	Municipal Police	New York	New York	April	51	Murder or Manslaughter	No	 Unknown	Unknown	0	Unkn
14996	14997	NY03030	New York	Municipal Police	New York	New York	April	52	Murder or Manslaughter	Yes	 Unknown	Male	19	В
14997	14998	NY03030	New York	Municipal Police	New York	New York	April	53	Murder or Manslaughter	Yes	 Unknown	Male	20	В
14998	14999	NY03030	New York	Municipal Police	New York	New York	April	54	Murder or Manslaughter	Yes	 Unknown	Female	27	В
14999	15000	NY03030	New York	Municipal Police	New York	New York	April	55	Murder or Manslaughter	Yes	 Unknown	Male	24	В
5000	rows x 2	3 columns												

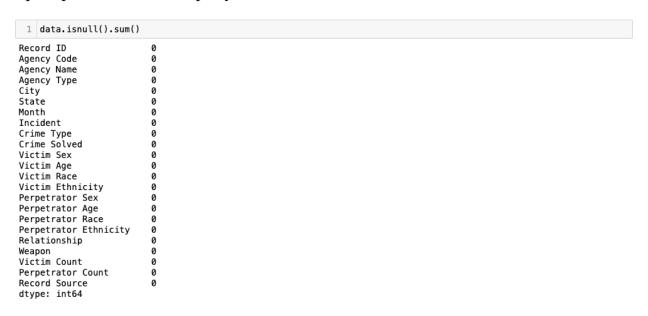
Список колонок с типами данных.

```
1 data.dtypes
Record ID
                                  int64
Agency Code
                                 object
Agency Name
                                 object
Agency Type
City
                                 object
object
State
                                 object
                                 object
int64
object
Month
Incident
Crime Type
Crime Solved
                                 object
Victim Sex
Victim Age
Victim Race
                                 object
                                 int64
object
Victim Ethnicity
                                 object
Perpetrator Sex
Perpetrator Age
Perpetrator Race
                                 object
                                 object
                                 object
Perpetrator Ethnicity
                                 object
Relationship
                                 object
Weapon
                                 object
Victim Count
                                   int64
Perpetrator Count
Record Source
dtype: object
                                 object
```

5 первых строк датасета.

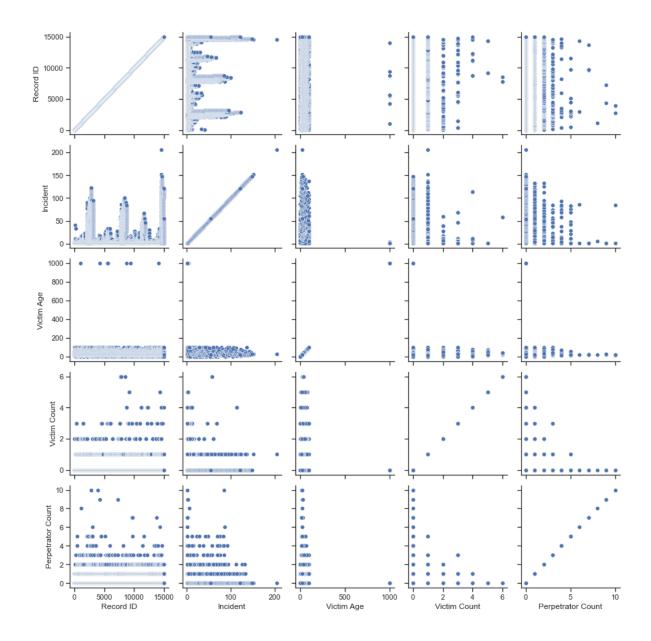


Проверка на наличие пропущеных значений



Создание графических представлений для более глубокого понимания информации.

```
1 sns.PairGrid(data).map(sns.scatterplot)
<seaborn.axisgrid.PairGrid at 0x18dbbe27a00>
```



4. Обработка данных

Кодирование категориальных признаков.

Масштабирование данных.

Для повышения эффективности алгоритмов машинного обучения выполним

нормализацию данных.

```
1 scale_cols = ['Record ID', 'Incident', 'Victim Age', 'Victim Count', 'Perpetrator Count']

1 sc1 = MinMaxScaler()
2 sc1_data = sc1.fit_transform(data_clone[scale_cols])
```

Включим в набор данных отмасштабированные значения.

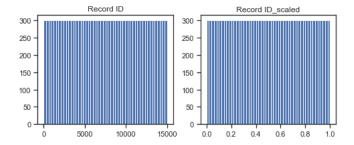
```
for i in range(len(scale_cols)):
    col = scale_cols[i]
    new_col_name = col + '_scaled'
    data_clone[new_col_name] = sc1_data[:,i]
```

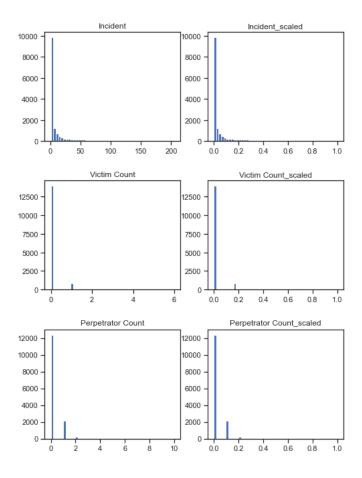
	Record ID	Agency Code	Agency Name	Agency Type	City	State	Month	Incident	Crime Type	Crime Solved	 Relationship	Weapon	Victim Count	Perpetrator Count		Record ID_scaled
0	1	AK00101	41	Municipal Police	17	Alaska	January	1	1	Yes	 Acquaintance	0	0	0	FBI	0.000000
1	2	AK00101	41	Municipal Police	17	Alaska	March	1	1	Yes	 Acquaintance	13	0	0	FBI	0.000067
2	3	AK00101	41	Municipal Police	17	Alaska	March	2	1	No	 Unknown	15	0	0	FBI	0.000133
3	4	AK00101	41	Municipal Police	17	Alaska	April	1	1	Yes	 Acquaintance	13	0	0	FBI	0.000200
4	5	AK00101	41	Municipal Police	17	Alaska	April	2	1	No	 Unknown	15	0	1	FBI	0.000267
14995	14996	NY03030	1084	Municipal Police	490	New York	April	51	1	No	 Unknown	8	0	0	FBI	0.999733
14996	14997	NY03030	1084	Municipal Police	490	New York	April	52	1	Yes	 Unknown	8	0	1	FBI	0.999800
14997	14998	NY03030	1084	Municipal Police	490	New York	April	53	1	Yes	 Acquaintance	15	0	0	FBI	0.999867
14998	14999	NY03030	1084	Municipal Police	490	New York	April	54	1	Yes	 Friend	9	0	0	FBI	0.999933
14999	15000	NY03030	1084	Municipal Police	490	New York	April	55	1	Yes	 Unknown	8	0	1	FBI	1.000000

Проверяем сохранение формы распределения данных после масштабирования.

```
for col in scale_cols:
    col_scaled = col + '_scaled'

fig, ax = plt.subplots(1, 2, figsize=(8,3))
    ax[0].hist(data_clone[col], 50)
    ax[1].hist(data_clone[col_scaled], 50)
    ax[0].title.set_text(col)
    ax[1].title.set_text(col_scaled)
    plt.show()
```





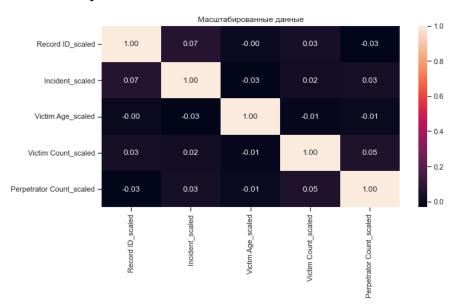
5. Анализ корреляции данных с последующим извлечением выводов относительно возможности построения моделей машинного обучения.

Исходные данные.



```
fig, ax = plt.subplots(figsize=(10,5))
sns.heatmap(data_clone[corr_cols_2].corr(), annot=True, fmt='.2f')
ax.set_title('Масштабированные данные')
plt.show()
```

Масштабированные данные.



6. Определение критериев оценки качества моделей для последующего использования.

Для оценки качества классификационной задачи выберем следующие метрики:

Сохранение и визуализация метрик.

Создадим класс для записи и отображения метрик эффективности построенных моделей.

```
1 class MetricLogger:
 3
           def __init__(self):
                __int__(set);
self.df = pd.DataFrame(
    {'metric': pd.Series([], dtype='str'),
    'alg': pd.Series([], dtype='str'),
    'value': pd.Series([], dtype='float')})
 4 5
 6
 9
           def add(self, metric, alg, value):
10
11
12
13
14
15
                 Добавление значения
                # Удаление значения если оно уже было ранее добавлено self.df.drop(self.df['metric']==metric)&(self.df['alg']==alg)].index, inplace = True)
                 # Добавление нового значения
                 temp = [{'metric':metric, 'alg':alg, 'value':value}]
self.df = pd.concat([self.df, pd.DataFrame(temp)], ignore_index=True)
16
17
18
19
20
21
22
23
24
25
26
27
28
           def get_data_for_metric(self, metric, ascending=True):
                 Формирование данных с фильтром по метрике
                 temp_data = self.df[self.df['metric']==metric]
                 temp_data_2 = temp_data.sort_values(by='value', ascending=ascending)
return temp_data_2['alg'].values, temp_data_2['value'].values
           def plot(self, str_header, metric, ascending=True, figsize=(5, 5)):
29
30
                 Вывод графика
                array_labels, array_metric = self.get_data_for_metric(metric, ascending)
fig, ax1 = plt.subplots(figsize=figsize)
31
32
33
                 pos = np.arange(len(array_metric))
34
                 rects = ax1.barh(pos, array_metric,
35
                                          align='center',
                                          height=0.5,
36
                tick_label=array_labels)
ax1.set_title(str_header)
for a,b in zip(pos, array_metric):
37
38
39
                      plt.text(0.5, a-0.05, str(round(b,3)), color='white')
40
41
```

7. Определение наилучших моделей для решения задачи классификации или регрессии.

Для решения задачи регрессии мы выберем следующие модели:

- Линейная регрессия
- Машина опорных векторов
- Решающее дерево
- Метод ближайших соседей
- Градиентный бустинг
- Случайный лес

8. Создание обучающего и тестового наборов данных из исходного датасета.

Используя отмасштабированные данные, разделим выборку на обучающий и тестовый наборы с помощью фильтрации.

Выборки для задачи классификации.

```
task_clas_cols = ["Record ID_scaled", "Victim Age_scaled", "Victim Count_scaled", "Perpetrator Count_scaled"]

X_train = data_clone[task_clas_cols]
X_test = data_clone[task_clas_cols]
Y_train = data_clone['Incident_scaled']
Y_test = data_clone['Incident_scaled']
X_train.shape, X_test.shape, Y_train.shape, Y_test.shape

((15000, 4), (15000, 4), (15000,), (15000,))
```

9. Создание начального решения (baseline) для выбранных моделей без настройки гиперпараметров. Модели обучаются на обучающем наборе данных, а затем оцениваются по качеству на тестовом наборе.

Решение задачи регрессии.

Модели.

Сохранение метрик.

```
def regr_train_model(model_name, model, regrMetricLogger):
    model.fit(X_train, Y_train)
    Y_pred = model.predict(X_test)

mae = mean_absolute_error(Y_test, Y_pred)
    mse = mean_squared_error(Y_test, Y_pred)
    r2 = r2_score(Y_test, Y_pred)

regrMetricLogger.add('MAE', model_name, mae)
    regrMetricLogger.add('MSE', model_name, mse)
    regrMetricLogger.add('MSE', model_name, r2)

print('{} \t MAE={}, MSE={}, R2={}'.format(
    model_name, round(mae, 3), round(r2, 3)))
```

10. Настройка гиперпараметров выбранных моделей с применением кросс-валидации.

```
n_{range} = np.array(range(1,1000,5))
     tuned_parameters = [{'n_neighbors': n_range}]
    tuned_parameters
[{'n_neighbors': array([ 1,
                                      6, 11, 16, 21, 26, 31, 36, 41, 46, 51, 56, 61, 86, 91, 96, 101, 106, 111, 116, 121, 126,
            66, 71,
                               81,
           131, 136, 141, 146, 151, 156, 161, 166, 171, 176, 181, 186, 191,
           196, 201, 206, 211, 216, 221, 226, 231, 236, 241, 246, 251,
           261, 266, 271, 276, 281, 286, 291, 296, 301, 306, 311, 316,
           326, 331, 336, 341, 346, 351, 356, 361, 366, 371, 376, 381,
           391, 396, 401, 406, 411, 416, 421, 426, 431, 436, 441, 446, 451,
           456, 461, 466, 471, 476, 481, 486, 491, 496, 501, 506, 511, 516,
           521, 526, 531, 536, 541, 546, 551, 556, 561, 566, 571, 576, 581,
           586, 591, 596, 601, 606, 611, 616, 621, 626, 631, 636, 641,
           651, 656, 661, 666, 671, 676, 681, 686, 691, 696, 701, 706, 711,
           716, 721, 726, 731, 736, 741, 746, 751, 756, 761, 766, 771, 776, 781, 786, 791, 796, 801, 806, 811, 816, 821, 826, 831, 836, 841,
                                                                    761,
           846, 851, 856, 861, 866, 871, 876, 881, 886, 891, 896, 901, 906, 911, 916, 921, 926, 931, 936, 941, 946, 951, 956, 961, 966, 971,
           976, 981, 986, 991, 996])}]
 1 %%time
 2 regr_gs = GridSearchCV(KNeighborsRegressor(), tuned_parameters, cv=5, scoring='neg_mean_squared_error')
 3 regr_gs.fit(X_train, Y_train)
CPU times: total: 5min 10s
GridSearchCV(cv=5, estimator=KNeighborsRegressor(),
        param_grid=[{'n_neighbors': array([ 1, 6, 11, 16, 21
66, 71, 76, 81, 86, 91, 96, 101, 106, 111, 116, 121, 126,
131, 136, 141, 146, 151, 156, 161, 166, 171, 176, 181, 186, 191,
                                                                          11, 16, 21,
                                                                                             26, 31, 36, 41, 46, 51, 56, 61,
        196, 201, 206, 211, 216, 221, 226, 231, 236, 241, 246, 251, 256, 261, 266, 271, 276, 281, 286, 291, 296, 301, 306, 311, 316, 321,
         326, 331, 336, 341, 346, 351, 356,...
         586, 591, 596, 601, 606, 611, 616, 621, 626, 631, 636, 641, 646,
         651, 656, 661, 666, 671, 676, 681, 686, 691, 696, 701, 706, 711,
         716, 721, 726, 731, 736, 741, 746, 751, 756, 761, 766, 771, 776, 781, 786, 791, 796, 801, 806, 811, 816, 821, 826, 831, 836, 841,
        846, 851, 856, 861, 866, 871, 876, 881, 886, 891, 896, 901, 906, 911, 916, 921, 926, 931, 936, 941, 946, 951, 956, 961, 966, 971,
         976, 981, 986, 991, 996])}],
                scoring='neg_mean_squared_error')
```

Лучшая модель.

```
1 regr_gs.best_estimator_

KNeighborsRegressor(n_neighbors=996)
```

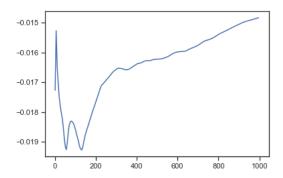
Лучшее значение параметров.

```
1 regr_gs.best_params_
{'n_neighbors': 996}

1 regr_gs_best_params_txt = str(regr_gs.best_params_['n_neighbors'])
2 regr_gs_best_params_txt
'996'
```

Изменение качества на тестовой выборке в зависимости от К-соседей.

```
plt.plot(n_range, regr_gs.cv_results_['mean_test_score'])
[<matplotlib.lines.Line2D at 0x18dd6755dc0>]
```



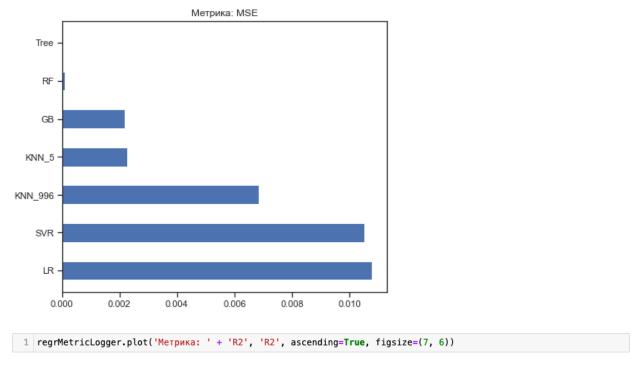
11.Применение оптимальных значений гиперпараметров для выбранных моделей и оценка их качества сравнительно с базовыми моделями.

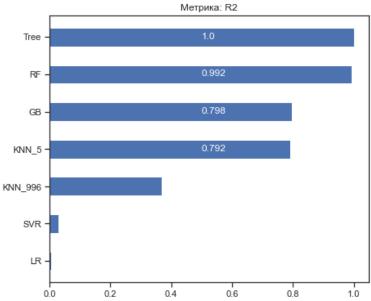
Сделаем выводы о производительности построенных моделей, основываясь на выбранных метриках.

Метрики качества модели.

```
regr_metrics = regrMetricLogger.df['metric'].unique()
 2 regr_metrics
array(['MAE', 'MSE', 'R2'], dtype=object)
 1 regrMetricLogger.plot('Метрика: ' + 'MAE', 'MAE', ascending=False, figsize=(7, 6))
                                  Метрика: МАЕ
    Tree
     RF
  KNN_5
KNN 996
     LR
    SVR
       0.00
              0.01
                     0.02
                             0.03
                                    0.04
                                           0.05
                                                  0.06
                                                         0.07
                                                                0.08
```

1 regrMetricLogger.plot('Метрика: ' + 'MSE', 'MSE', ascending=False, figsize=(7, 6))





12.Вывод

С учетом двух из трех использованных метрик, наилучшей моделью оказалась та, которая базируется на методе опорных векторов (SVR).

Заключение

В рамках нашего научного исследования мы изучили набор данных "История преступлений США" с помощью различных инструментов анализа данных, таких как Pandas, Matplotlib, Seaborn, Numpy и Streamlit, используя язык программирования Python в Jupyter Notebook.

Мы начали с анализа структуры данных, заполнили пропущенные значения, определили признаки для построения моделей, провели кодирование категориальных признаков, выполнили масштабирование данных и создали дополнительные признаки для улучшения качества моделей.

Проведен корреляционный анализ данных, который позволил сделать промежуточные выводы о возможности построения моделей машинного обучения.

Для оценки качества моделей были выбраны метрики, включая точность, полноту, F1-меру и ROC AUC. Мы рассмотрели более пяти различных моделей, в том числе базовые и ансамблевые методы.

Сформированы обучающая и тестовая выборки, построены базовые решения для выбранных моделей, а также подобраны оптимальные значения гиперпараметров с использованием методов кросс-валидации.

В итоге проведена оценка качества построенных моделей и их сравнение с базовыми моделями, что позволило сделать выводы о наилучших подходах к решению задачи классификации или регрессии, основываясь на выбранных метриках.

Список использованных источников информации

- 1) Абдрахманов, М. И. Devpractice Team. Pandas. Pабота с данными. / М. И. Абдрахманов 2-е изд. devpractice.ru. 2020. 170 с.: [Электронный ресурс]. // URL: https://coderbooks.ru/books/python/pandas_rabota_s_dannymi_abdrahmano v 2020/ (дата обращения: 15.03.2024)
- 2) Абдрахманов, М. И. Devpractice Team. Python. Визуализация данных. Matplotlib. Seaborn. Mayavi. devpractice.ru. 2020. 412 с.: [Электронный ресурс]. // URL: https://coderbooks.ru/books/python/python_vizualizaciya_dannyh_abdrahm anov 2020/ (дата обращения: 21.03.2024)
- 3) Методические указания по программному обеспечению «Pandas»: [Электронный ресурс]. // URL: https://pandas.pydata.org/ (дата обращения: 20.02.2024)
- 4) Методические указания по программному обеспечению «Seaborn»: [Электронный ресурс]. // URL: https://seaborn.pydata.org/ (дата обращения: 22.02.2024)
- 5) Метрики качества. Метрики классификации. Подсчет количества ошибок, доли правильных ответов, точности, полноты. [Электронный ресурс]. // URL: https://data-scientists.ru/metriki-kachestva-metriki-klassifikacii-podschet-kolichestva-oshibok-doli-pravilnyh-otvetov (дата обращения: 01.03.2024)
- 6) Регрессионные модели в Python: [Электронный ресурс]. // URL: https://nagornyy.me/it/regressionnye-modeli-v-python/ (дата обращения: 11.03.2024)
- 7) Учебник по машинному обучению: [Электронный ресурс]. // URL: https://education.yandex.ru/handbook/ml (дата обращения: 01.03.2024)