

Практическое задание №1. Метрические алгоритмы классификации

Петренко Дарья, 317 группа

24 октября 2018 г.

Содержание

Введение	2
Список экспериментов	2
Эксперимент №1. Сравнение времени работы алгоритмов на разных подмножествах признаков.	2
Эксперимент №2. Сравнение точности и времени работы метода с различными параметрами по кросс-валидации	3
Эксперимент №3. Сравнение взвешенного метода с методом без весов.	4
Эксперимент №4. Анализ работы алгоритма с подобранными параметрами	4
Эксперимент №5. Размножение обучающей выборки	6
Повороты изображений	7
Смещение по каждой из двух размерностей в каждую из 2-х сторон	8
Дисперсия фильтра Гаусса	9
Комбинации преобразований	10
Эксперимент №6. Размножение тестовой выборки	12
Повороты изображений	13
Смещения	14
Дисперсия фильтра Гаусса	15
Комбинации преобразований	18

Введение

В ходе выполнения практического задания была написана собственная реализация алгоритма поиска ближайших соседей, а также проведен ряд экспериментов, в ходе которых исследовались как собственная реализация, так и библиотечные функции, реализующие этот метод. Исследовались время работы алгоритма и точность предсказаний для различных значений параметров, а также исследовалось, как влияет на точность размножение тестовой и обучающей выборок.

Список экспериментов

Для проведения экспериментов был загружен датасет MNIST - база данных образцов рукописного написания цифр. Загруженный при помощи функции *fetch_mldata* библиотеки *sklearn.datasets*, датасет представляет из себя матрицу объектов - изображений рукописных цифр от 0 до 9 (каждое изображение представлено вектором из 764 элементов) и вектор целевых переменных - их числовых представлений. При этом база данных делится на обучающую выборку (первые 60 тыс объектов) и тестовую (последние 10 тыс).

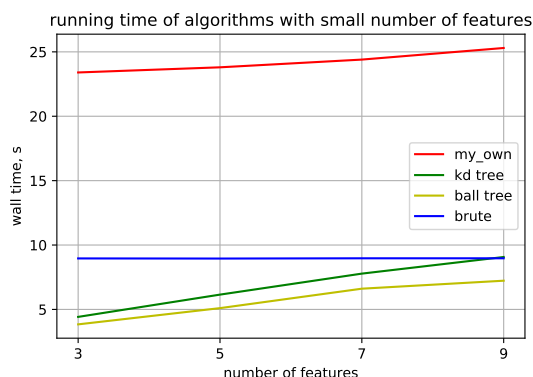
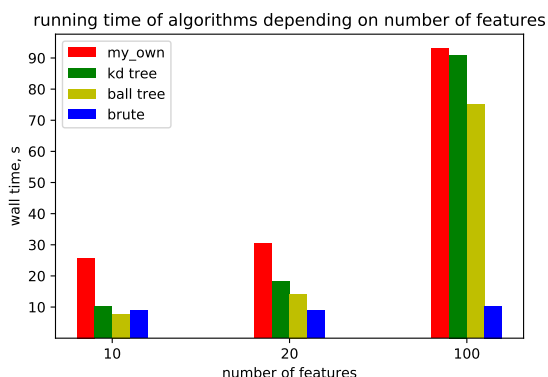
Эксперимент №1. Сравнение времени работы алгоритмов на разных подмножествах признаков.

Для проведения эксперимента были случайным образом выбраны подмножества признаков, по которым считается расстояние, размера 10, 20 и 100. Для каждого из поддерживаемых моделью алгоритмов ('my_own', 'brute', 'kd_tree', 'ball_tree') был реализован поиск 5-ти ближайших соседей без весов для евклидовой метрики при *test_block_size* = 1000 и измерено время работы алгоритма. Результаты продемонстрированы на графике слева.

Можно заметить, что методы, основанные на построении деревьев, показывают хорошие результаты только на пространствах малых размерностей, и при большом числе признаков наиболее эффективным по времени работы является алгоритм 'brute'. При этом алгоритм 'ball_tree' работает быстрее алгоритма 'kd_tree'. Алгоритм 'my own' в силу примитивности реализации показывает худшие результаты на всех тестах.

Для более точной демонстрации этого факта был проведен дополнительный эксперимент: было измерено время работы алгоритмов на пространствах малых размерностей (график справа). Можно заметить, что на пространстве признаков размерности 3 алгоритмы 'kd_tree' и 'ball_tree' работают более чем в два раза быстрее алгоритма 'brute', но с увеличением числа признаков их время работы стремительно растет, в то время как время работы алгоритма 'brute' почти не меняется. Как и в первом эксперименте, алгоритм 'ball_tree' работает быстрее 'kd_tree', а алгоритм 'my_own' показывает худший результат с большим отрывом от остальных.

Можно предположить, что алгоритмы, основанные на построении деревьев, эффективны по времени только на пространствах малых размерностей, тк при построении дерева происходит разбиение пространства на два полупространства по каждому измерению, и при $n \ll 2^d$ (n - количество объектов в обучающей выборке, d - размерность пространства признаков) большая часть листьев дерева не содержит объектов выборки.

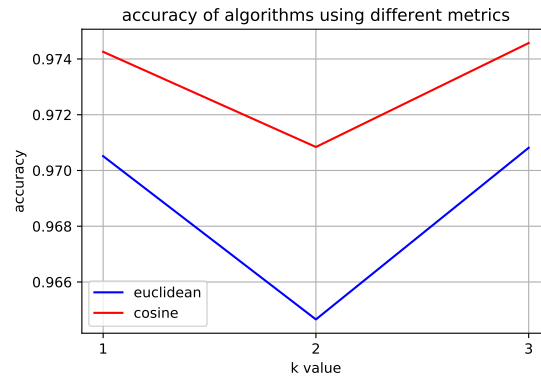
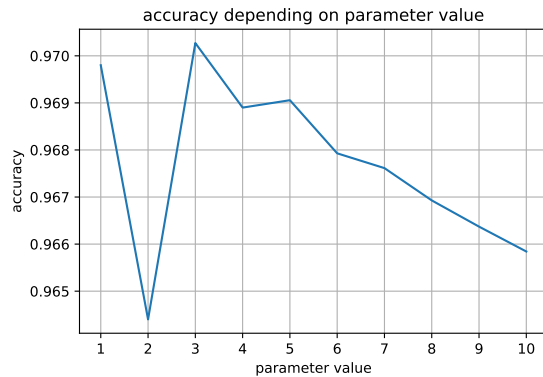


Эксперимент №2. Сравнение точности и времени работы метода с различными параметрами по кросс-валидации

По результатам эксперимента №1 было выяснено, что на пространстве признаков большой размерности наиболее эффективным по времени является алгоритм 'brute', поэтому именно он использовался при проведении эксперимента №2. Эксперимент проводился на полном наборе признаков, был реализован поиск без весов.

В первой части эксперимента была измерена точность предсказания в зависимости от параметра k (число соседей). Использовалась евклидова метрика. Результаты эксперимента представлены на графике слева. Можно видеть, что оптимальным значением параметра k является $k=3$, близкая к нему точность достигается при $k=1$. При $k=2$ точность значительно меньше (резкий скачок на графике), а при $k > 3$ она постепенно снижается с увеличением значения параметра.

Во второй части эксперимента было измерено время работы, а также точность предсказания на кросс-валидации для евклидовой и косинусной метрики. В качестве значения параметра k были взяты те, при которых достигалась наибольшая точность в первой части эксперимента, а именно $k=1$ и $k=3$. Результаты измерения точности представлены на графике слева. Алгоритм, использующий косинусную метрику, работает дольше (кросс-валидация для $k = [1, 3]$ заняла 4 минуты 53 секунды для евклидовой и 6 минут 13 секунд для косинусной метрики), но показывает лучшую точность для обоих значений параметра.



Эксперимент №3. Сравнение взвешенного метода с методом без весов.

Эксперимент был проведен для параметров, которые оказались наиболее эффективными по результатам предыдущих экспериментов: $k=3$, косинусная метрика, алгоритм *'brute'*. Была измерена точность на кросс-валидации для взвешенного метода и метода без весов, а также измерено время работы.

Время работы почти одинаково: 3 минуты 37 секунд для взвешенного метода и 3 минуты 36 секунд для метода без весов. Точность взвешенного метода выше и равна 0.9747, в то время как точность метода без весов - 0.9739.

Эксперимент №4. Анализ работы алгоритма с подобранными параметрами

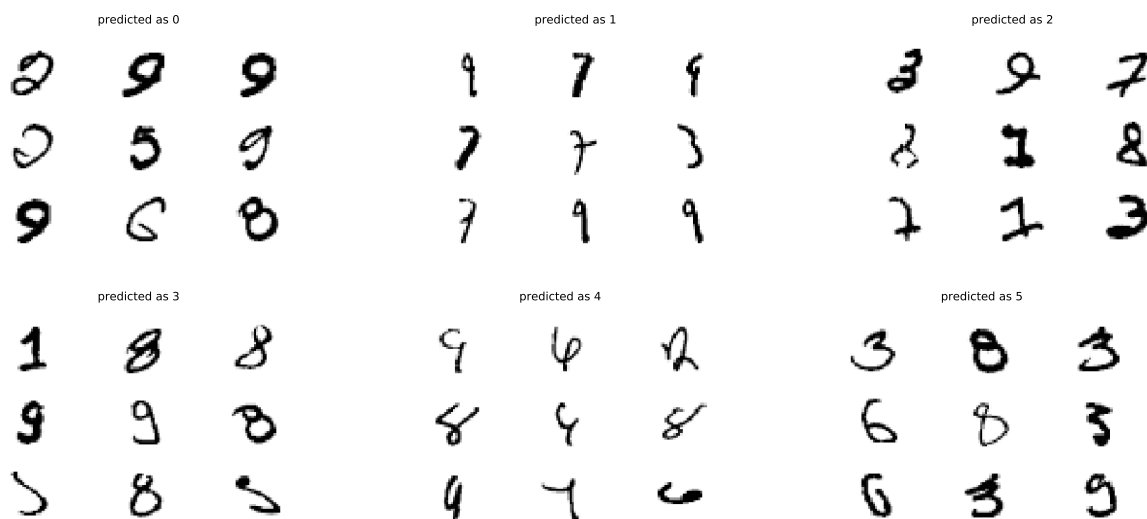
Лучшая точность по результатам предыдущего эксперимента достигается для значений параметров: $k=3$, косинусная метрика, алгоритм *'brute'*, используются веса. В ходе эксперимента метод с такими параметрами был применен к обучающей и тестовой выборке, была измерена точность предсказания. Она равна 0.9742. Точность по кросс-валидации составляет 0.9747. Небольшая разница между ними может свидетельствовать об отсутствии переобучения.

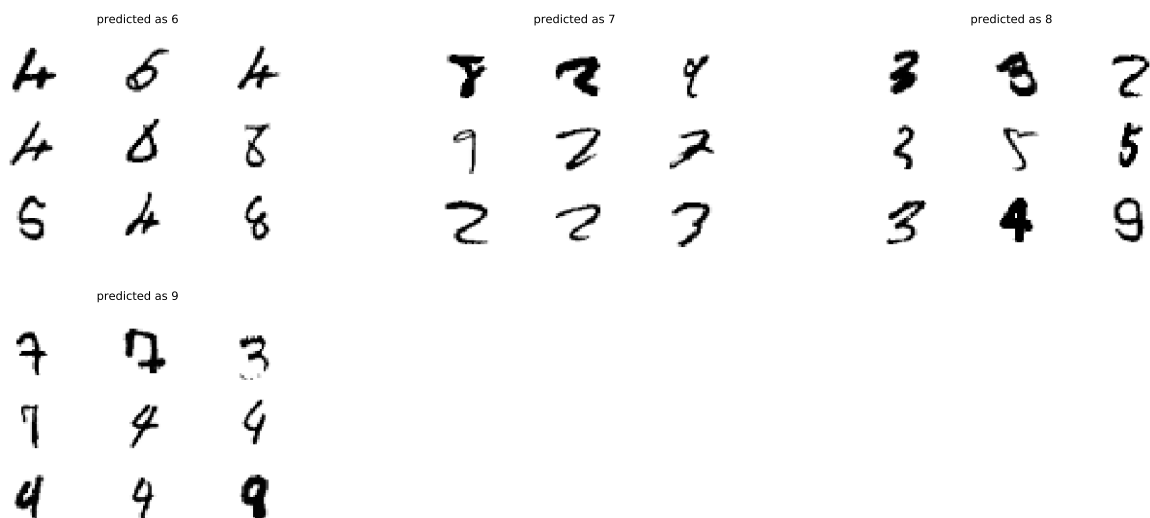
Задача распознавания рукописных цифр из датасета MNIST была предложена для решения на Kaggle, и первые 31 лидирующих алгоритмов добились точности 1.00000.

С помощью функции *confusion_matrix* из библиотеки *scikit-learn* была построена матрица ошибок.

$$\begin{bmatrix} 977 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1130 & 3 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 8 & 1 & 1009 & 1 & 1 & 0 & 0 & 8 & 4 & 0 \\ 0 & 1 & 3 & 975 & 0 & 13 & 0 & 5 & 7 & 6 \\ 2 & 2 & 0 & 0 & 946 & 0 & 6 & 1 & 1 & 24 \\ 4 & 0 & 0 & 9 & 1 & 859 & 7 & 1 & 6 & 5 \\ 4 & 2 & 0 & 0 & 2 & 3 & 947 & 0 & 0 & 0 \\ 1 & 10 & 6 & 0 & 1 & 0 & 0 & 996 & 0 & 14 \\ 6 & 2 & 2 & 10 & 4 & 4 & 4 & 3 & 936 & 3 \\ 8 & 7 & 2 & 5 & 7 & 3 & 1 & 6 & 3 & 967 \end{bmatrix}$$

Можно заметить, что ошибочно предсказанные и реальные цифры похожи или имеют схожие элементы. Например, за 0 чаще всего принимались цифры, имеющие одну ярко выраженную круглую часть (чаще всего 2, 8, 9); за 1 - цифры, имеющие ярко выраженную вертикальную линию (7, 9); за 2 - цифры с горизонтальной чертой внизу или полукруглым изгибом сверху (1, 3, 7); за 3 - цифры с одним или двумя полукруглыми изгибами справа (5, 8, 9); за 4 - цифры с полукругом сверху и вертикальной чертой внизу (9); за 5 - цифры с характерным изгибом влево (3, 8, 9); за 6 - цифры с окружностью внизу и наклонной чертой сверху (4, 5, 8); за 7 - цифры с горизонтальной чертой сверху и наклонной внизу (2, 3, 9); за 8 - цифры с двумя окружностями (2, 3, 5), за 9 - цифры с окружностью сверху и наклонной чертой внизу (3, 4, 7). Это подтверждается матрицей ошибок и приведенными ниже примерами объектов, при распознавании которых были допущены ошибки. Как можно заметить, большинство изображенных цифр имеют отсутствующие или искаженные части, что затрудняет распознавание.





Эксперимент №5. Размножение обучающей выборки

Для проведения эксперимента было сформировано разбиение обучающей выборки для кросс-валидации с 3-мя фолдами, оно было единым для всех тестов.

Была написана вспомогательная функция, которая для каждого из 3-х разбиений кросс-валидации применяет рассматриваемое преобразование ко всем объектам из обучающей выборки и находит среди них ближайших соседей для всех объектов из валидационной выборки (используется метод *KNNClassifier* с оптимальными параметрами, найденными по результатам предыдущих экспериментов). Формируются матрицы индексов в обучающей выборке и расстояний до ближайших соседей; функция возвращает списки таких матриц для каждого из разбиений.

Для нахождения ближайших соседей среди нескольких преобразованных матриц из соответствующих списков извлекаются матрицы индексов и расстояний. Матрицы конкатенируются, путем сортировки по расстоянию до исходного объекта находятся индексы ближайших соседей в валидационной выборке (исходной или преобразованной). Так как преобразования инвариантны относительно класса объекта, то по индексу в любой из выборок (исходной или преобразованной) можно однозначно определить класс найденного соседа. Зная классы ближайших соседей и расстояния до них, предсказываем класс объекта из валидационной выборки.

Повороты изображений

Результаты описанного эксперимента для поворотов изображений (повороты производились в каждую из двух сторон):

Зависимость точности от угла поворота

углы поворотов	точность предсказания
15	0.9784
10	0.9809
5	0.9802

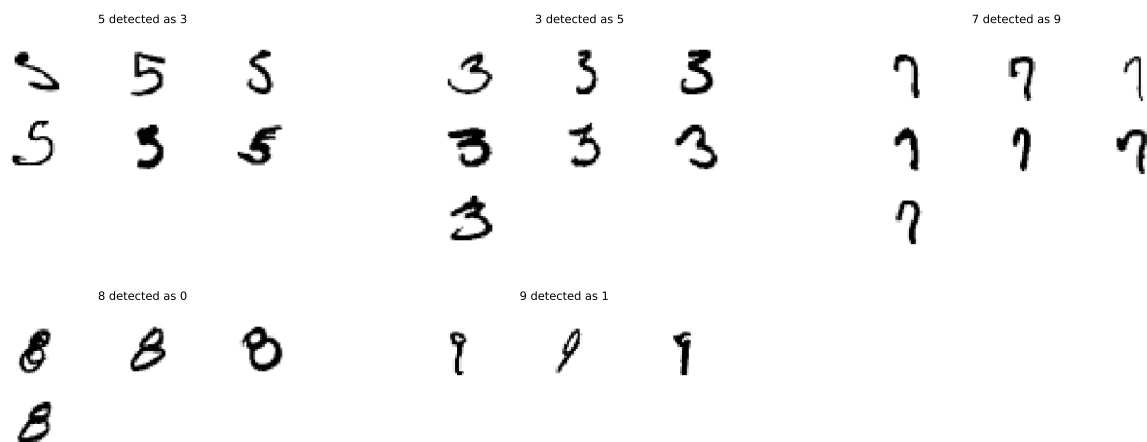
Максимальная точность на кросс-валидации была достигнута при поворотах на 10 градусов, применим это преобразование ко всей обучающей выборке и построим матрицу ошибок. Полученная точность составляет 0.981.

$$\begin{bmatrix} 977 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1132 & 2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 6 & 2 & 1009 & 2 & 1 & 0 & 0 & 10 & 2 & 0 \\ 0 & 0 & 1 & 986 & 1 & 6 & 0 & 4 & 8 & 4 \\ 1 & 1 & 1 & 0 & 949 & 0 & 5 & 2 & 0 & 23 \\ 2 & 0 & 0 & 6 & 1 & 873 & 3 & 3 & 2 & 2 \\ 2 & 3 & 0 & 0 & 0 & 1 & 951 & 0 & 1 & 0 \\ 2 & 7 & 3 & 0 & 0 & 0 & 0 & 1008 & 0 & 8 \\ 2 & 0 & 1 & 7 & 4 & 2 & 3 & 3 & 947 & 5 \\ 4 & 4 & 1 & 6 & 5 & 1 & 1 & 6 & 3 & 978 \end{bmatrix}$$

Для упрощения анализа выведем поэлементную разность матриц ошибок преобразованной выборки и исходной.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & -1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 \\ -2 & 1 & 0 & 1 & 0 & 0 & 0 & 2 & -2 & 0 \\ 0 & -1 & -2 & 11 & 1 & -7 & 0 & -1 & 1 & -2 \\ -1 & -1 & 1 & 0 & 3 & 0 & -1 & 1 & -1 & -1 \\ -2 & 0 & 0 & -3 & 0 & 14 & -4 & 2 & -4 & -3 \\ -2 & 1 & 0 & 0 & -2 & -2 & 4 & 0 & 1 & 0 \\ 1 & -3 & -3 & 0 & -1 & 0 & 0 & 12 & 0 & -6 \\ -4 & -2 & -1 & -3 & 0 & -2 & -1 & 0 & 11 & 2 \\ -4 & -3 & -1 & 1 & -2 & -2 & 0 & 0 & 0 & 11 \end{bmatrix}$$

Как и предполагалось, уменьшилось количество ошибок в распознавании цифр, зависящих от степени наклона или содержащих наклонные элементы: 3, 5, 7, 9, 1, 8. Выведем несколько объектов, которые после преобразования выборки стали распознаваться корректно.



Смещение по каждой из двух размерностей в каждую из 2-х сторон

Результаты эксперимента для операции смещения (на кросс-валидации с 3-мя фолдами):

Зависимость точности от смещения

смещение	точность предсказания
на 1 пиксель по одной размерности (4 направления)	0.9817
на 2 пикселя по одной размерности (4 направления)	0.9775
на 3 пикселя по одной размерности (4 направления)	0.9755
на 1 пиксель по обеим размерностям (4 комбинации)	0.9808

Максимальная точность на кросс-валидации была достигнута при смещении на 1 пиксель по одной размерности. Применим это преобразование ко всей обучающей выборке и построим матрицу ошибок. Точность на тестовой выборке составляет 0.9805.

977	0	0	0	0	0	2	1	0	0
0	1132	3	0	0	0	0	0	0	0
7	0	1014	0	0	0	1	9	1	0
1	1	1	986	0	8	0	5	5	3
1	1	0	0	953	0	6	0	1	20
3	0	0	10	1	864	6	2	3	3
3	2	0	0	0	1	952	0	0	0
1	7	4	0	2	0	0	1004	0	10
3	1	2	8	1	4	4	3	945	3
6	3	1	3	4	4	0	9	1	978

Поэлементная разность матриц ошибок преобразованной выборки и исходной:

$$\begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & -1 & 5 & -1 & -1 & 0 & 1 & 1 & -3 & 0 \\ 1 & 0 & -2 & 11 & 0 & -5 & 0 & 0 & -2 & -3 \\ -1 & -1 & 0 & 0 & 7 & 0 & 0 & -1 & 0 & -4 \\ -1 & 0 & 0 & 1 & 0 & 5 & -1 & 1 & -3 & -2 \\ -1 & 0 & 0 & 0 & -2 & -2 & 5 & 0 & 0 & 0 \\ 0 & -3 & -2 & 0 & 1 & 0 & 0 & 8 & 0 & -4 \\ -3 & -1 & 0 & -2 & -3 & 0 & 0 & 0 & 9 & 0 \\ -2 & -4 & -1 & -2 & -3 & 1 & -1 & 3 & -2 & 11 \end{bmatrix}$$

У объектов, которые алгоритм начал правильно распознавать после применения преобразования к обучающей выборке, не наблюдается сдвигов. Как видно из разности матриц ошибок, улучшения встречаются во всех классах объектов в примерно равных количествах.

Дисперсия фильтра Гаусса

Результаты эксперимента на кросс-валидации с 3-мя фолдами:

Зависимость точности от параметра σ

значение параметра	точность предсказания
0.5	0.9763
1	0.9802
1.5	0.9785

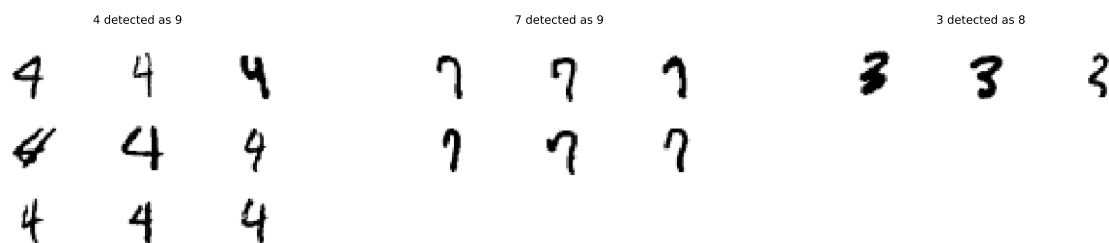
Максимальная точность на кросс-валидации была достигнута при значении параметра $\sigma = 1$. Применим это преобразование ко всей обучающей выборке и построим матрицу ошибок. Точность на тестовой выборке составляет 0.9805.

$$\begin{bmatrix} 977 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1131 & 3 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 7 & 1 & 1004 & 2 & 0 & 0 & 2 & 13 & 3 & 0 \\ 0 & 0 & 2 & 987 & 1 & 8 & 0 & 4 & 3 & 5 \\ 0 & 0 & 0 & 0 & 960 & 0 & 4 & 2 & 0 & 16 \\ 2 & 1 & 0 & 8 & 1 & 869 & 4 & 1 & 2 & 4 \\ 4 & 3 & 0 & 0 & 1 & 3 & 947 & 0 & 0 & 0 \\ 0 & 8 & 5 & 0 & 3 & 1 & 1 & 1001 & 0 & 9 \\ 3 & 0 & 1 & 4 & 3 & 5 & 2 & 4 & 949 & 3 \\ 3 & 4 & 0 & 1 & 9 & 3 & 1 & 6 & 2 & 980 \end{bmatrix}$$

Поэлементная разность матриц ошибок до и после преобразования обучающей выборки.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -5 & 1 & -1 & 0 & 2 & 5 & -1 & 0 \\ 0 & -1 & -1 & 12 & 1 & -5 & 0 & -1 & -4 & -1 \\ 2 & -2 & 0 & 0 & 14 & 0 & -2 & 1 & -1 & -8 \\ 2 & 1 & 0 & -1 & 0 & 10 & -3 & 0 & -4 & -1 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & -2 & -1 & 0 & 2 & 1 & 1 & 5 & 0 & -5 \\ -3 & -2 & -1 & -6 & -1 & 1 & -2 & 1 & 13 & 0 \\ -5 & -3 & -2 & -4 & 2 & 0 & 0 & 0 & -1 & 13 \end{bmatrix}$$

Уменьшилось число ошибок в распознавании цифр, которые были написаны нечетко, имели утолщенные или не прописанные до конца линии. Это и ожидалось при сглаживании изображения. Выведем несколько объектов, которые после преобразования выборки стали распознаваться корректно.



Комбинации преобразований

Будем рассматривать комбинации преобразований со значениями параметров, показавшими самую высокую точность по результатам предыдущих экспериментов, то есть повороты на 10 градусов, смещения на 1 пиксель и наложение фильтра Гаусса со значением сигма=1. (Любой набор помимо перечисленных преобразованных матриц включает исходную обучающую выборку)

Измерим точность на кросс-валидации с 3-мя фолдами.

Зависимость точности от комбинаций преобразований

комбинация преобразований	точность предсказания
по одной матрице для каждого из преобразований (всего 8 шт)	0.9846
по одной матрице для всех, кроме фильтра Гаусса (всего 7 шт)	0.9834
по одной матрице для всех, кроме вращений (всего 6 шт)	0.9829
по одной матрице для всех, кроме сдвигов (всего 4 шт)	0.9824
матрицы с вращениями, на которые наложен фильтр Гаусса + матрицы со сдвигами	0.9844
матрицы с вращениями, на которые наложен фильтр Гаусса + матрицы со сдвигами + матрица с фильтром Гаусса	0.9850
матрицы с вращениями, на которые наложен фильтр Гаусса + матрицы со сдвигами, на которые наложен фильтр Гаусса	0.98592
матрицы с вращениями, на которые наложен фильтр Гаусса + матрицы со сдвигами, на которые наложен фильтр Гаусса + матрица с фильтром Гаусса	0.98585
8 различных комбинаций сдвигов и вращений, на которые наложен фильтр Гаусса	0.9848
8 различных комбинаций сдвигов и вращений, на которые наложен фильтр Гаусса + матрица с фильтром Гаусса	0.9855

Таким образом, из всех рассмотренных комбинаций наибольшая точность достигается на наборе матриц с вращениями и матриц со сдвигами, к каждой из которых применен фильтр Гаусса. Применим эти преобразования ко всей обучающей выборке. Получили точность 0.9865. Это больше, чем на кросс-валидации, что свидетельствует о том, что модель не переобучена. Построим матрицу ошибок.

$$\begin{bmatrix} 979 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1133 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 1 & 1015 & 1 & 0 & 0 & 0 & 9 & 1 & 0 \\ 1 & 0 & 1 & 993 & 1 & 4 & 0 & 3 & 5 & 2 \\ 0 & 1 & 0 & 0 & 962 & 0 & 3 & 1 & 0 & 15 \\ 2 & 0 & 0 & 6 & 0 & 873 & 6 & 1 & 2 & 2 \\ 2 & 2 & 0 & 0 & 1 & 1 & 952 & 0 & 0 & 0 \\ 0 & 6 & 4 & 0 & 1 & 0 & 0 & 1012 & 0 & 5 \\ 1 & 0 & 2 & 4 & 2 & 3 & 1 & 3 & 955 & 3 \\ 1 & 4 & 0 & 0 & 4 & 2 & 0 & 5 & 2 & 991 \end{bmatrix}$$

Поэлементная разность матриц ошибок до преобразований и после:

$$\begin{bmatrix} 2 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 3 & -1 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\ -3 & 0 & 6 & 0 & -1 & 0 & 0 & 1 & -3 & 0 \\ 1 & -1 & -2 & 18 & 1 & -9 & 0 & -2 & -2 & -4 \\ -2 & -1 & 0 & 0 & 16 & 0 & -3 & 0 & -1 & -9 \\ -2 & 0 & 0 & -3 & -1 & 14 & -1 & 0 & -4 & -3 \\ -2 & 0 & 0 & 0 & -1 & -2 & 5 & 0 & 0 & 0 \\ -1 & -4 & -2 & 0 & 0 & 0 & 0 & 16 & 0 & -9 \\ -5 & -2 & 0 & -6 & -2 & -1 & -3 & 0 & 19 & 0 \\ -7 & -3 & -2 & -5 & -3 & -1 & -1 & -1 & -1 & 24 \end{bmatrix}$$

Число неверных предсказаний во всех классах снизилось, большим остается лишь число ошибочного предсказания 9 вместо 4 (согласно матрице ошибок). Немного снизить это количество можно, добавив в матрицам преобразований матрицу, на которую наложен фильтр Гаусса. Общая точность предсказания на всей тестовой выборке снизится до 0.9864, но матрица ошибок будет выглядеть следующим образом:

$$\begin{bmatrix} 979 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1133 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 1 & 1013 & 0 & 1 & 0 & 1 & 11 & 1 & 0 \\ 1 & 0 & 1 & 991 & 1 & 6 & 0 & 3 & 5 & 2 \\ 0 & 0 & 0 & 0 & 966 & 0 & 3 & 1 & 0 & 12 \\ 1 & 0 & 0 & 6 & 0 & 876 & 6 & 1 & 1 & 1 \\ 2 & 2 & 0 & 0 & 1 & 1 & 952 & 0 & 0 & 0 \\ 0 & 6 & 5 & 0 & 2 & 0 & 0 & 1008 & 0 & 7 \\ 1 & 0 & 2 & 3 & 2 & 3 & 2 & 3 & 955 & 3 \\ 1 & 3 & 0 & 0 & 6 & 1 & 0 & 5 & 2 & 991 \end{bmatrix}$$

Эксперимент №6. Размножение тестовой выборки

Используем кросс-валидацию с 3-мя фолдами для подбора параметров. Каждому объекту валидационной выборки сопоставим множество, состоящее из объектов, полученных путем применения рассматриваемых преобразований к исходному. Расстоянием от множества до объекта обучающей выборки будем считать минимум расстояния до этого объекта по всем элементам множества.

Таким образом, для нахождения k ближайших соседей каждого из объектов валидационной выборки достаточно найти k соседей для каждого из его преобразований, а затем из полученных пар выбрать k объектов с наименьшим расстоянием.

Повороты изображений

Результаты проведенных экспериментов представлены в таблице.

Зависимость точности от угла поворота

угол поворота	точность предсказания
5	0.9781
10	0.9794
15	0.9773

Наибольшая точность достигается при поворотах на 10 градусов. Применим эти преобразования ко всей тестовой выборке. Достигнутая точность составляет 0.9791.

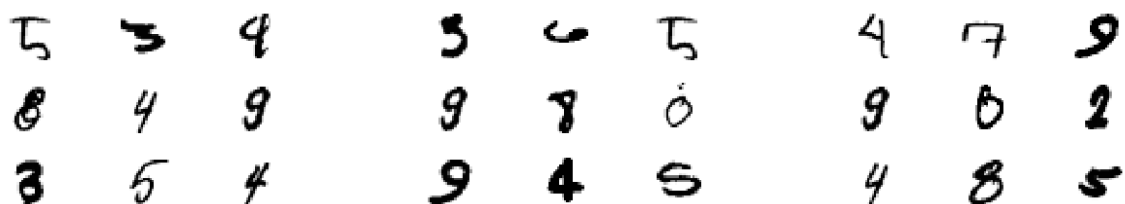
Построим матрицу ошибок.

$$\begin{bmatrix} 977 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 0 & 0 \\ 0 & 1133 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6 & 1 & 1009 & 1 & 1 & 0 & 1 & 11 & 2 & 0 \\ 0 & 0 & 3 & 984 & 1 & 7 & 0 & 4 & 7 & 4 \\ 0 & 2 & 1 & 0 & 947 & 0 & 7 & 3 & 0 & 22 \\ 3 & 0 & 0 & 5 & 1 & 867 & 6 & 2 & 4 & 4 \\ 3 & 3 & 0 & 0 & 1 & 0 & 950 & 0 & 1 & 0 \\ 2 & 7 & 3 & 0 & 0 & 0 & 0 & 1006 & 0 & 10 \\ 3 & 0 & 3 & 7 & 3 & 2 & 2 & 3 & 946 & 5 \\ 4 & 4 & 2 & 5 & 7 & 5 & 1 & 5 & 4 & 972 \end{bmatrix}$$

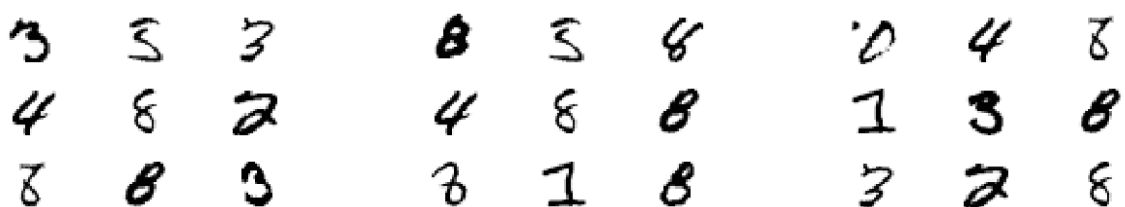
Выведем поэлементную разность матриц ошибок для поворотов на 10 градусов обучающей и тестовой выборки.

$$\begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 1 & 0 & 0 & -1 & 0 \\ -1 & 1 & 0 & 0 & -2 & 0 & 2 & 1 & 0 & -1 \\ 1 & 0 & 0 & -1 & 0 & -6 & 3 & -1 & 2 & 2 \\ 1 & 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 2 \\ 1 & 0 & 2 & 0 & -1 & 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 & 2 & 4 & 0 & -1 & 1 & -6 \end{bmatrix}$$

Можно увидеть, что почти для всех классов объектов предсказания стали хуже, тк на диагонали стоят преимущественно отрицательные числа, а вне диагоналей - преимущественно положительные. Выведем примеры объектов, классы которых были правильно предсказаны при преобразовании обучающей выборки и неправильно - при преобразовании тестовой.



Примеры объектов, классы которых, наоборот, были правильно предсказаны при преобразовании тестовой выборки и неправильно - при преобразовании обучающей.



Как видно из матрицы ошибок, повороты тестовой выборки исправляют те же типы ошибок, что и повороты обучающей.

Смещения

Результаты эксперимента для операции смещения (по кросс-валидации с 3-мя фолдами).

Зависимость точности от смещения

смещение	точность предсказания
на 1 пиксель по одной размерности (4 направления)	0.9817
на 2 пикселя по одной размерности (4 направления)	0.9770
на 3 пикселя по одной размерности (4 направления)	0.9747
на 1 пиксель по двум размерностям (4 комбинации)	0.9799

Наибольшая точность достигнута при смещении на 1 пиксель в одном из 4-х направлений. Применим это преобразование ко всей тестовой выборке и выведем матрицу ошибок. Полученная точность составляет 0.9805, что совпадает с точностью, достигнутой при применении аналогичного преобразования к обучающей выборке. Более того, из матрицы ошибок равны, а векторы предсказанных значений целевой переменной полностью совпадают. Таким образом, для операции смещения применения преобразований к тестовой и обучающей выборкам полностью равносильны.

Дисперсия фильтра Гаусса

Результаты эксперимента (по кросс-валидации с 3-мя фолдами).

Зависимость точности от значения параметра σ

значение параметра	точность предсказания
0.5	0.9744
1	0.9611
1.5	0.9216

Наибольшая точность достигается при значении параметра $\sigma = 0.5$. Применим это преобразование ко всей тестовой выборке и построим матрицу ошибок. Достигнутая точность составляет 0.9725, она меньше точности, достигаемой на тестовой выборке без применения преобразований.

$$\begin{bmatrix} 978 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1129 & 3 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 10 & 0 & 1001 & 2 & 1 & 0 & 2 & 9 & 7 & 0 \\ 0 & 1 & 2 & 979 & 0 & 10 & 0 & 5 & 9 & 4 \\ 2 & 1 & 0 & 0 & 942 & 0 & 6 & 1 & 1 & 29 \\ 3 & 1 & 0 & 12 & 1 & 851 & 8 & 1 & 8 & 7 \\ 4 & 2 & 0 & 0 & 2 & 3 & 947 & 0 & 0 & 0 \\ 3 & 10 & 7 & 0 & 2 & 0 & 0 & 994 & 0 & 12 \\ 4 & 3 & 2 & 11 & 3 & 3 & 6 & 4 & 936 & 2 \\ 7 & 8 & 1 & 3 & 8 & 3 & 1 & 7 & 3 & 968 \end{bmatrix}$$

Поэлементная разность матриц ошибок для преобразованной тестовой выборки и исходной.

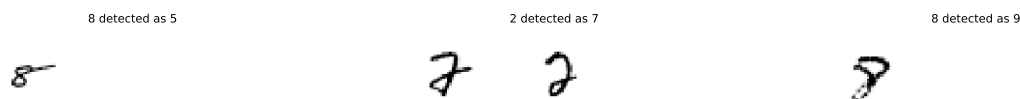
$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 2 & -1 & -8 & 1 & 0 & 0 & 2 & 1 & 3 & 0 \\ 0 & 0 & -1 & 4 & 0 & -3 & 0 & 0 & 2 & -2 \\ 0 & -1 & 0 & 0 & -4 & 0 & 0 & 0 & 0 & 5 \\ -1 & 1 & 0 & 3 & 0 & -8 & 1 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 1 & 0 & 0 & -2 & 0 & -2 \\ -2 & 1 & 0 & 1 & -1 & -1 & 2 & 1 & 0 & -1 \\ -1 & 1 & -1 & -2 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Преимущественно отрицательные числа на диагонали и положительные вне диагонали подтверждают факт того, что точность предсказаний снизилась по сравнению с моделью без размножения выборки.

Поэлементная разность матриц ошибок для моделей с размножением тестовой выборки и аналогичным размножением обучающей (с оптимальными параметрами для каждого случая: $\sigma = 0.5$ для преобразования тестовой выборки и $\sigma = 1$ для преобразования обучающей).

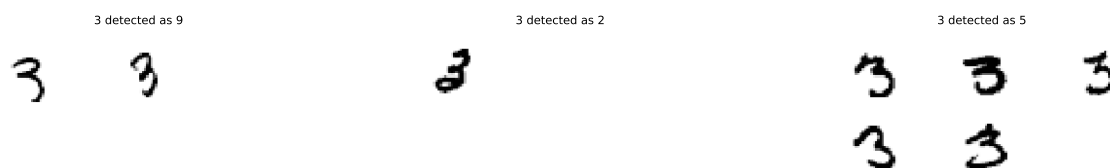
$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -2 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 3 & -1 & -3 & 0 & 1 & 0 & 0 & -4 & 4 & 0 \\ 0 & 1 & 0 & -8 & -1 & 2 & 0 & 1 & 6 & -1 \\ 2 & 1 & 0 & 0 & -18 & 0 & 2 & -1 & 1 & 13 \\ 1 & 0 & 0 & 4 & 0 & -18 & 4 & 0 & 6 & 3 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 3 & 2 & 2 & 0 & -1 & -1 & -1 & -7 & 0 & 3 \\ 1 & 3 & 1 & 7 & 0 & -2 & 4 & 0 & -13 & -1 \\ 4 & 4 & 1 & 2 & -1 & 0 & 0 & 1 & 1 & -12 \end{bmatrix}$$

Примеры объектов, классы которых были неправильно предсказаны при преобразовании обучающей выборки и правильно - при преобразовании тестовой.



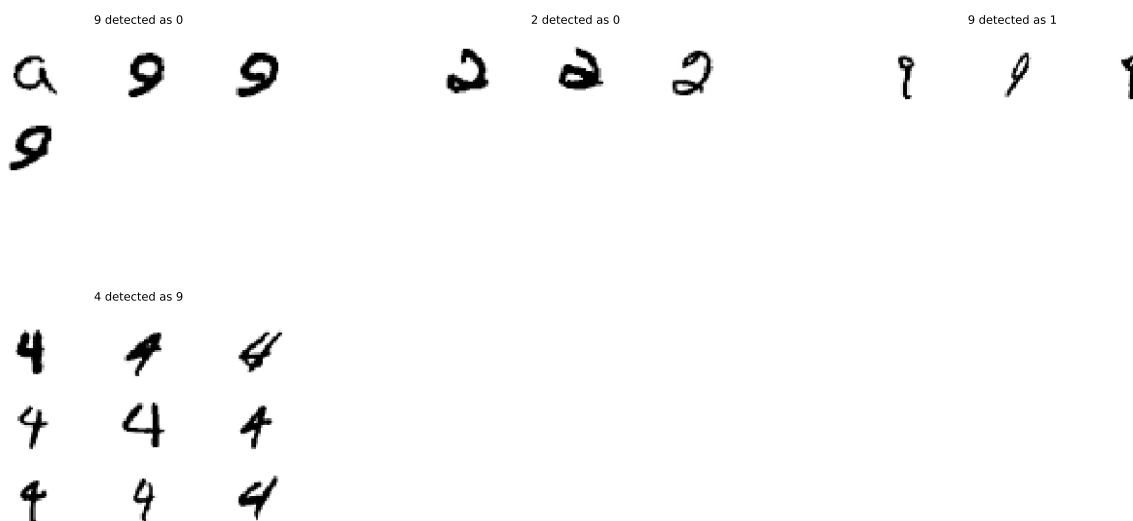
Можно сделать вывод, что применение фильтра Гаусса к тестовой выборке помогает лучше распознавать цифры, которые содержат лишние штрихи, пропадающие при размытии (например, наклонный штрих сверху у цифры 8 из первого примера, из-за которого она ошибочно распознавалась как 5, или полукруг снизу у цифры 9 из третьего примера, из-за которого она ошибочно распознавалась как 9).

Примеры объектов, классы которых были неправильно предсказаны при использовании модели без размножения выборок и правильно - при преобразовании тестовой.



Приведенные примеры подтверждают приведенное выше предположение: применение фильтра помогает сгладить лишний элемент в нижней части цифры 3 из второго примера, из-за которого она ошибочно распознавалась как 2.

Примеры объектов, классы которых были неправильно предсказаны при преобразовании тестовой выборки и правильно - при преобразовании обучающей.



Можно сделать вывод, что, в отличие от применения фильтра Гаусса к тестовой выборке, применение его к обучающей помогает правильно распознавать цифры, изображенные слишком жирными линиями.

Таким образом, несмотря на то, что применение фильтра Гаусса к тестовой выборке помогает лучше предсказывать классы некоторых объектов, их относительное количество очень мало, и многие другие объекты, наоборот, распознаются неверно. Общая точность предсказания такой модели падает по сравнению с моделью, использующей применение фильтра к обучающей выборке и моделью, не использующей размножение выборок.

Комбинации преобразований

Тк в ходе экспериментов было выяснено, что применение фильтра Гаусса к тестовой выборке приводит к ухудшению качества модели, то будем рассматривать только комбинации поворотов и смещений с найденными оптимальными параметрами (смещение на 1 пиксель и поворот на 10 градусов). Результаты на кросс-валидации с 3-мя фолдами представлены в таблице.

Зависимость точности от комбинации преобразований

комбинация преобразований	точность предсказания
4 матрицы только со сдвигами + 2 матрицы только с вращениями	0.9813
8 различных комбинаций сдвигов и вращений	0.9812
8 различных комбинаций сдвигов и вращений + 4 матрицы только со сдвигами	0.9829
8 различных комбинаций сдвигов и вращений + 2 матрицы только с вращениями	0.9812
8 различных комбинаций сдвигов и вращений + 4 матрицы только со сдвигами + 2 матрицы только с вращениями	0.9825

Помимо перечисленных преобразований в поиске соседей всегда участвует и исходная тестовая выборка.

Наибольшая точность достигается, если использовать все возможные комбинации поворотов и смещений и отдельно матрицы только со смещениями. Применим эти преобразования ко всей тестовой выборке. Достигнутая точность составляет 0.9818. Построим матрицу ошибок.

$$\begin{bmatrix} 977 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 0 & 0 \\ 0 & 1132 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 5 & 2 & 1016 & 0 & 0 & 0 & 1 & 6 & 2 & 0 \\ 1 & 0 & 3 & 991 & 0 & 2 & 0 & 3 & 7 & 3 \\ 0 & 2 & 0 & 0 & 951 & 0 & 4 & 1 & 0 & 24 \\ 3 & 0 & 0 & 5 & 1 & 867 & 10 & 2 & 3 & 1 \\ 4 & 3 & 0 & 0 & 0 & 1 & 950 & 0 & 0 & 0 \\ 2 & 5 & 3 & 0 & 1 & 0 & 0 & 1008 & 0 & 9 \\ 2 & 1 & 2 & 4 & 2 & 4 & 4 & 3 & 950 & 2 \\ 6 & 4 & 0 & 4 & 6 & 5 & 0 & 7 & 1 & 976 \end{bmatrix}$$

Поэлементная разность между матрицами ошибок модели с размножением тестовой выборки и без.

$$\begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 2 & -1 & -1 & 0 & 0 & -1 & 0 & 0 & 1 \\ -3 & 1 & 7 & -1 & -1 & 0 & 1 & -2 & -2 & 0 \\ 1 & -1 & 0 & 16 & 0 & -11 & 0 & -2 & 0 & -3 \\ -2 & 0 & 0 & 0 & 5 & 0 & -2 & 0 & -1 & 0 \\ -1 & 0 & 0 & -4 & 0 & 8 & 3 & 1 & -3 & -4 \\ 0 & 1 & 0 & 0 & -2 & -2 & 3 & 0 & 0 & 0 \\ 1 & -5 & -3 & 0 & 0 & 0 & 0 & 12 & 0 & -5 \\ -4 & -1 & 0 & -6 & -2 & 0 & 0 & 0 & 14 & -1 \\ -2 & -3 & -2 & -1 & -1 & 2 & -1 & 1 & -2 & 9 \end{bmatrix}$$