Федеральное государственное образовательное

бюджетное учреждение высшего образования

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ**

**ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

ИНСТИТУТ ОКРЫТОГО ОБРАЗОВАНИЯ

**Департамент анализа данных и машинного обучения**

Пояснительная записка к курсовой работе

по дисциплине “Технологии анализа данных и машинное обучение”

на тему:

**“Машинное обучение в задачах распознавания объектов на фотографии”**

Направление подготовки - 09.03.09 «Прикладная информатика»

**Выполнил студент:**

4-го курса группы ЗБ-ПИ20-1

Дервук М.А.

**Научный руковотель:**

Добрина Мария Валерьевна

Москва – 2024 г

Содержание

[Введение 3](#_Toc167724528)

[1. Подготовительный этап 5](#_Toc167724529)

[1.1. Архитектура нейронной сети 6](#_Toc167724530)

[1.2. Подготовка выборки из базы данных 6](#_Toc167724531)

[1.3. Начало разработки программы 7](#_Toc167724532)

[1.4. Процесс машинного обучения моделей 9](#_Toc167724533)

[1.4.1. Метод ближайших соседей 9](#_Toc167724534)

[1.4.2. Метод C-опорных векторов 11](#_Toc167724535)

[1.4.3. Метод K-средних. 12](#_Toc167724536)

[1.4.4. Метод Гауссовского наивного байесовского алгоритма. 13](#_Toc167724537)

[1.4.5. Метод случайного леса. 13](#_Toc167724538)

[1.4.6. Метод опорных векторов линейного ядра. 14](#_Toc167724544)

[2. Анализ данных 15](#_Toc167724545)

[2.1. Метод К-ближайших соседей. 15](#_Toc167724546)

[2.2. Метод С-опорных векторов 19](#_Toc167724547)

[2.3. Метод К-средних 22](#_Toc167724548)

[2.4. Метод Гауссовского наивного байесовского алгоритма 26](#_Toc167724549)

[2.5. Метод случайного леса 30](#_Toc167724550)

[2.6. Метод С-опорных векторов линейного ядра 34](#_Toc167724551)

[2.7. Общая оценка моделей 37](#_Toc167724552)

[ЗАКЛЮЧЕНИЕ 39](#_Toc167724553)

[Источники 41](#_Toc167724554)

# **Введение**

Машинное обучение в задачах распознавания объектов на изображении является **актуальным** для пользователей, так как программные средства такой технологии совершенствуют задачи пользователей в аспектах ускорения процессов задач пользователей и безопасности. К примеру, так оценивают единый государственный экзамен, либо работает биометрия или поиск по картинкам. Машинное обучение может происходит с учителем и без него. Прежде всего, распознавание объектов на изображении – это задачи классификации и сегментации. В данной программе внимание уделено больше методам классификации и машинному обучению с учителем. Перед программами для распознавания объектов на изображении с помощью машинного обучения стоят много проблем, вот некоторые из них:

1. Неполная база данных;
2. Некорректная база данных;
3. Отсутствие базы данных;
4. Ошибка в методе при реализации продукта под задачи;
5. Некорректные объекты и соответственно низкая оценка соответствия после обучения модели;
6. Нехватка системных ресурсов для обучения модели.

Решение данных проблем в машинном обучении при реализации программ для распознавания объектов на изображении является актуальным.

**Целью разработки** ПО является создание программы, которая:

* Производит машинное обучение моделей на базе данных и сравнивает методы обучения моделей по результатам эффективности.

Для достижения цели приложение выполнит **задачи**:

* Выбрать методы для моделей машинного обучения;
* Выбрать базу данных для машинного обучения;
* Разработать программу с методами моделей;
* Произвести оценку методов машинного обучения.

# **Подготовительный этап**

Для разработки программы для распознавания объектов на изображении путём машинного обучения определены следующие требования (табл.1) [1]:

Таблица 1. Требования к программе

|  |  |
| --- | --- |
| Замеры времени | В целях анализа временной сложности алгоритмов. Все инструкции, запускающие цикл обучения модели должны содержать замер времени обучения. Замер можно производить с помощью магических инструкций Jupyter или (более предпочтительно) с использованием стандартной библиотеки Python. Сравнение моделей должно учитывать и время обучения. |
| Визуализация | Программа должна визуализировать информацию. Особенно на этапах описательного анализа и анализа обучаемости модели. Оценивается разнообразие, наглядность и информативность визуализации. |
| Разнообразие моделей | Программа должна работать с разнообразными моделями обучения, применимыми к одной задаче. |
| Предобработка данных | Программа должна содержать исчерпывающий алгоритм предварительной обработки данных. Он служит для того, чтобы исправить все несовершенства в данных и сделать набор данных как можно более пригодным для машинного обучения. |
| Использование метрик эффективности | Программа должна быть разнообразной и адекватной задаче примененных метрик эффективности (включая время обучения) а также полнота сравнения и правильность выводов из сравнения моделей по разным метрикам. |
| Валидность результатов | Программа должна оценивать достоверность измерения метрик моделей. |

Критерием качества оценки результата разработки программы является реализация следующей задачи:

- Успешное обучение моделей путем машинного обучения и сравнения результатов оценки каждого метода.

## **Архитектура нейронной сети**

Нейронная сеть программы распознавания объектов на изображении является задачей классификации. Архитектура нашей нейронной сети состоит из:

* Входного слоя;
* Скрытого слоя;
* Выходного слоя.

На входном слое поступающее на вход растровое изображение преобразуется из двумерного объекта в одномерный, путем обозначения числового номера оттенка в градиенте цвета. Затем двумерный числовой массив преобразуется последовательно в одномерный.

На скрытом слое происходит машинное обучение.

Выходной слой выдаёт предсказание к какому типу объекта относится распознанный объект.

## **Подготовка выборки из базы данных**

В настоящее время существуют множество языков программирования, которые используются для разработки программ для распознавания объектов на изображении: Python, R, Java, Julia и LISP. Возможно использование и других языков, на которых можно вести машинное обучение разработки методов моделей в программах для распознавания объектов на изображении, но все они предполагают поддержку модулей и библиотек, которые работают значительно быстрее, чем создание аналогов. Практическая часть курсовой работы написана на языке Python.

При запуске программы пользователь должен дождаться окончания машинного обучения моделей, которые распознают объекты на изображении. После завершения машинного обучения модели выводится результат соответствия и изображение результатов работы модели на объектах базы данных. После этого выводятся графики.

После запуска программы, пользователю не будут доступны функции управления программой.

База данных для машинного обучения называется digit-recognizer и взята с сайта <https://www.kaggle.com/competitions/digit-recognizer>.

## **Начало разработки программы**

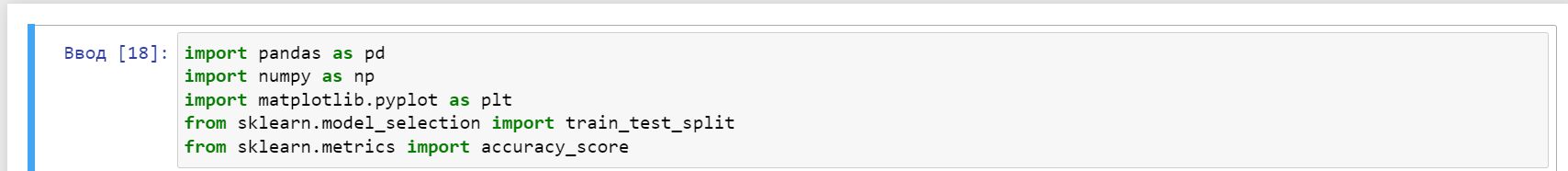
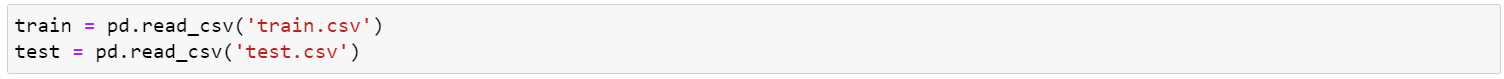
Для начала импортируем библиотеки [Рисунок 1.3.1.].

Рисунок 1.3.1. Импорт библиотек.

На рисунке изображены:

* библиотека pandas для работы с крупными файлами базы данных формата csv;
* Математическая библиотека numpy;
* Библиотека matplotlib, модуль pyplot – для вывода таблиц, диаграмм и изображений;
* Из библиотеки sklearn, модуля model\_selection загружаем функцию train\_test\_split для осуществления подготовки данных к машинному обучению моделей;
* Из библиотеки sklearn, модуля metrics, загружаем функцию accuracy\_score для комплексной оценки модели в рамках соответствия. Данная оценка покажет, насколько качество работает выбранный метод модели на выборке базы данных.

Далее загружаем базу данных, у нас она разделена на csv файлы для обучения и теста: “train.csv”, “test.csv”. Показано на рисунке 1.3.2.

Рисунок 1.3.2. Загрузка датасета

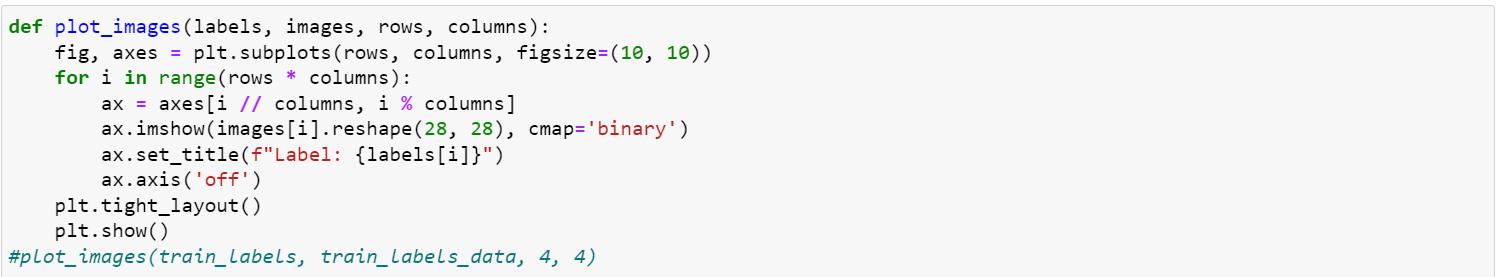
Загружаем в переменные train и test базы данных для машинного обучения модели и для тестирования обученной модели.

Далее, нам следует разделить датасет по признакам и по атрибутам признаков. Проводим данную операцию в коде на рисунке 1.3.3.

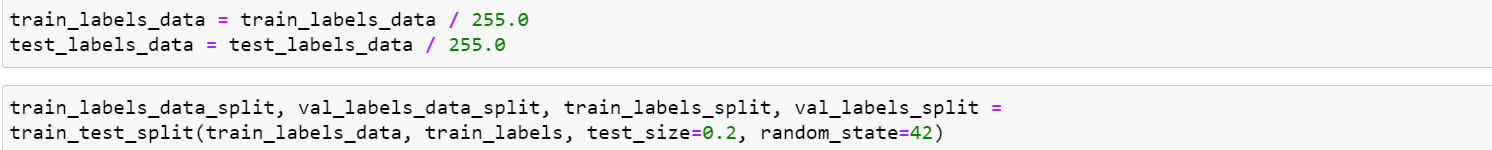
Рисунок 1.3.3. Форматирование данных выборки.

Здесь происходит следующее: переменная train\_labels принимает данные в виде признаков, переменная train\_labels\_data принимает значение атрибутов признаков предыдущей переменной, а test\_labels\_data содержит атрибуты признаков, которые мы должны определить после обучения моделей.

Так как нам придётся проверять выполненную работу моделей, заранее пишем функцию plot\_images, которая будет выводить 16 изображений для первичной оценки. В функции вывода изображений мы их сжимаем в размерность 28 28. Функция показана на рисунке 1.3.4.

Рисунок 1.3.4. Функция вывода изображений.

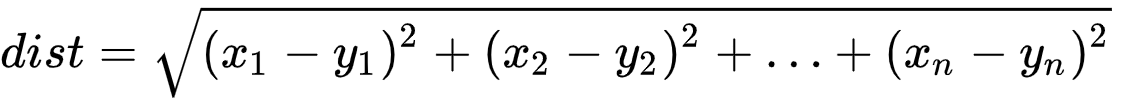
Рассмотрим, что такое атрибут признака. Это пиксель целого изображения, размерность которого в свою очередь составляет 256 256. Так как изображения даны в черно-белом градиенте, то содержимое одного пикселя – это численное представление в векторной шкале оттенков от 0 до 255. Следовательно, для удобства и ускорения процесса машинного обучения делим переменные, которые содержат атрибуты признаков числа на 255 и получаем числа от 0 до 1. Следующим действием подготавливаем данные путём разделения их на тренировочные и данные для валидации c помощью функции train\_test\_plit на рисунке 1.3.5, где мы оставляет пятую часть на тестирование.

Рисунок 1.3.5. Конечная подготовка данных для обучения моделей.

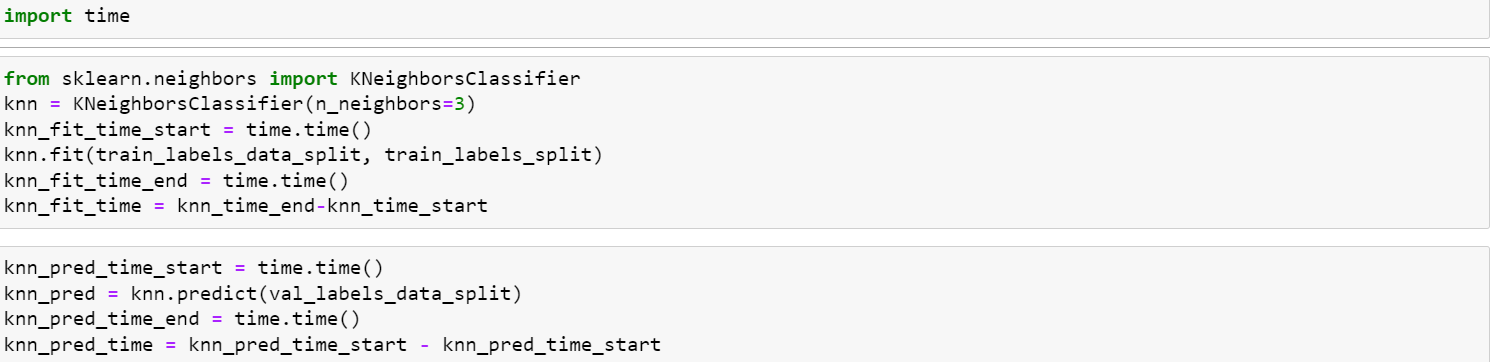
## **Процесс машинного обучения моделей**

### Метод ближайших соседей

По умолчанию функция использует метрику Минковского. Математическое представление данной функции на рисунке 1.4.1.1.

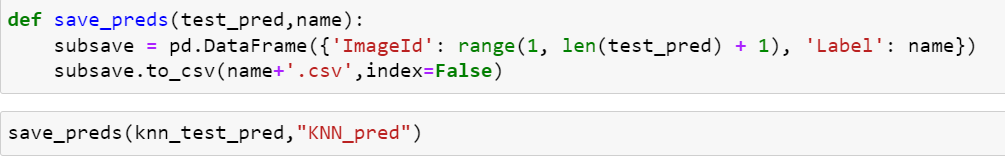
Рисунок 1.4.1.1. Математическое представление алгоритма ближайших соседей.

Функцией fit, на рисунке 1.4.1.2., тренируем модель, которой обозначали 3 ближайших соседа.

Рисунок 1.4.1.2. Обучение модели методом ближайших соседей.

Далее производим классификацию и предсказываем результат фунцией predict в переменную knn\_pred, следом за этим производим то же самое на тестовой выборке[Рисунок 1.4.1.3.].

Рисунок 1.4.1.3. Предсказывание обученной knn тестовой выборки.

Полученные результаты сохраняем в csv файл функцией save\_preds на рисунке 1.4.1.4.Рисунок 1.4.1.4. Функция сохранения результата предсказывания knn.

Далее импортируем функцию классификации и функцию матрицы корреляции: classification\_report и confusion\_matrix на рисунке 2.1.4.

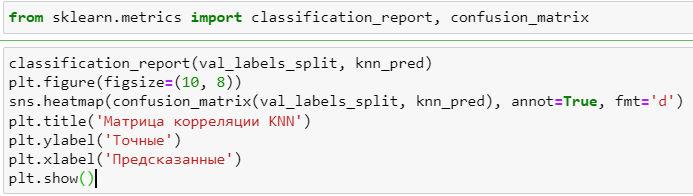
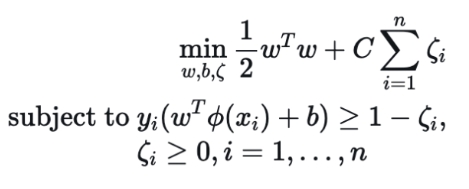


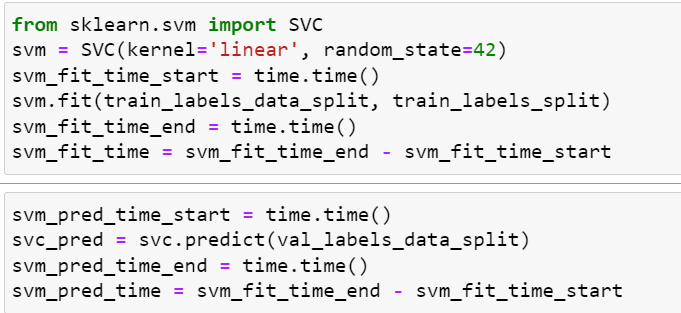
Рисунок 1.4.1.5. Код реализации матрицы

### Метод C-опорных векторов

Математическая формула данного метода на рисунке 1.4.2.1. Представляет собой линейно неразделимую выборку. Математическая формула говорит, что нужно отнять от отступа в плоскости положительную величину, но введенные поправки должны быть минимальны.

Рисунок 1.4.2.1. Математическое представление метода С-опорных векторов.

Далее, импортируем из библиотеки sklearn функцию SVC модуля SVM на рисунке 1.4.2.2.

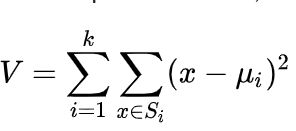
Рисунок 1.4.2.2. Тренировка и предсказывание метода C-опорных векторов.

Также, как и с предыдущей моделью предсказываем результат на тестовой выборке функцией predict на рисунке 1.4.2.3.

Рисунок 1.4.2.3. Предсказание на тестовой выборке svc.

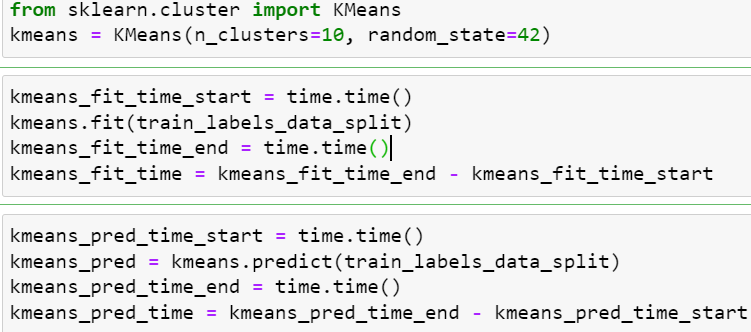
### Метод K-средних.

Данный метод больше используется для задач кластеризации. Математически метод представляет собой и минимизирует суммарное квадратичное отклонение точек кластеров от центров других кластеров. Суммарное количество кластеров умножается на центры масс всех векторов x из кластера S и умножается на квадратичную разницу векторов минус центра массы вектора S.

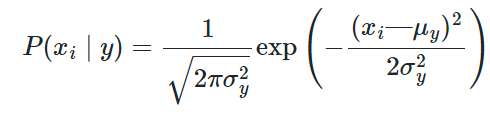
Рисунок 1.4.3.1. Математическое представление метода К-средних.

Далее импортируем из библиотеки sklearn функцию KMeans модуля cluster, ставим количество кластеров равно 10, случайно распределяя массы.

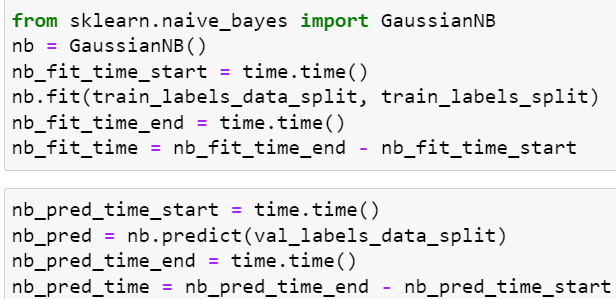
Производим тренировку и предсказание результатов на тренировочных выборках на рисунке 1.4.3.2.

Рисунок 1.4.3.2. Код метода KMeans.

### Метод Гауссовского наивного байесовского алгоритма.

Математическое представление Гауссовского наивного байесовского на рисунке 1.4.4.1. Где апостериорная вероятность принадлежности образца P равна априорной вероятности принадлежности случайного наблюдения умноженной на правдоподобие признаков x в классе, деленная на вероятность признаков x. Рисунок 1.4.4.1. Математическая формула Гауссовского наивного байесовского метода.

Нам нужна функция GaussianNB библиотеки sklearn, модуля naive\_bayes. Далее тренируем и предсказываем на проверочной выборке на рисунке 1.4.4.1.

Рисунок 1.4.4.1. Математическая формула гауссовского наивного Байеса.

### Метод случайного леса.

Математическая формула данного алгоритма на рисунке 1.4.5.1. Это простое древо принятия решений.Рисунок 1.4.5.1. Математическая формула метода случайного леса

Импортируем функцию RandomForestClassifier из библиотеки sklearn, добавляем 100 деревьев или ветвей, производим машинное обучение модели данным методом и проверяем на проверочных данных. Код указан на рисунке 1.4.5.2.

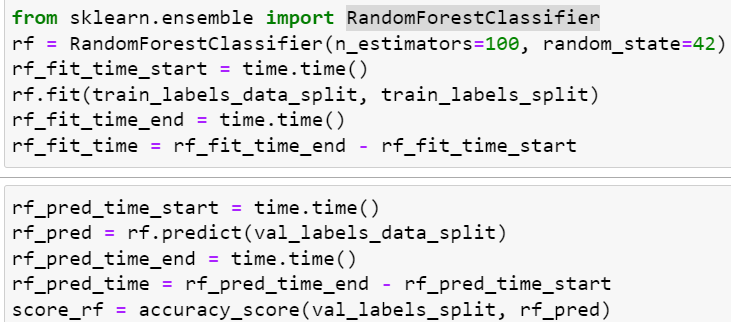
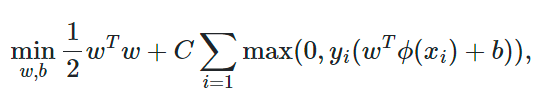


Рисунок 1.4.5.2. Код модуля метода случайных деревьев.

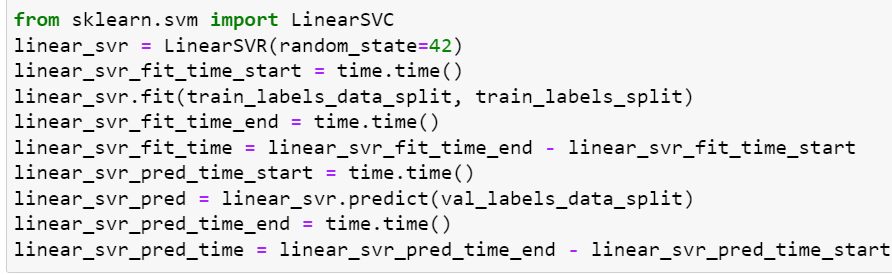


### Метод опорных векторов линейного ядра.

Математическая формула данного алгоритма на рисунке 1.4.6.1.

Рисунок 1.4.6.1. Математическая формула LinearSVC

Загружаем из sklearn.svm функцию LinearSVC, обучаем модель и тестируем

Рисунок 1.4.6.2. Код метода LinearSVC.

Следующая часть кода выводит графики рассеяния, распределения и гистограммы. Код на рисунке 1.4.6.3.

Рисунок 1.4.6.3. Код вывода графика рассеяния, графика распределения и гистограммы.

# **Анализ данных**

## **Метод К-ближайших соседей.**

На рисунке 2.1.1. Функция вывода изображений knn.



Рисунок 2.1.1. Вывод knn изображений.

Результат вывода результата можно увидеть на рисунке 2.1.2. Судя по тому, что признак «label» совпадает с атрибутами признака – это говорит о высокой оценке соотствия выполнения классификации моделью.

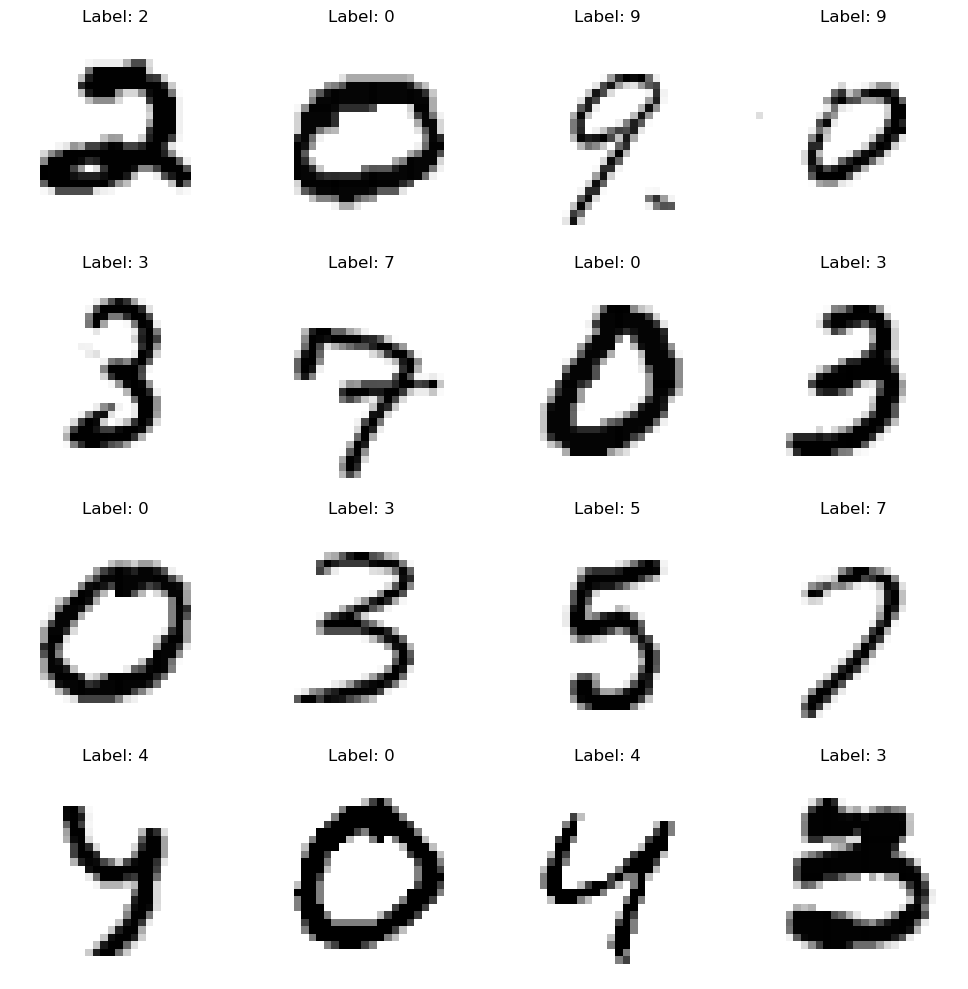


Рисунок 2.1.2. Изображения knn.

Произведем оценку соответствия полученным результатам тестируя выборку с помощью функции accuracy\_score на рисунке 2.1.3.:

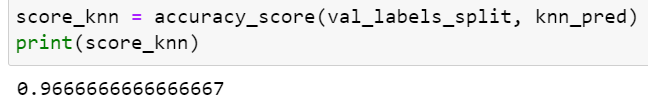


Рисунок 2.1.3. Вывод оценки knn.

Верно, 96% это отличный результат.

Далее создаём матрицу корреляции, чтобы увидеть картину сортировки изображений по признакам на рисунке 2.1.4.

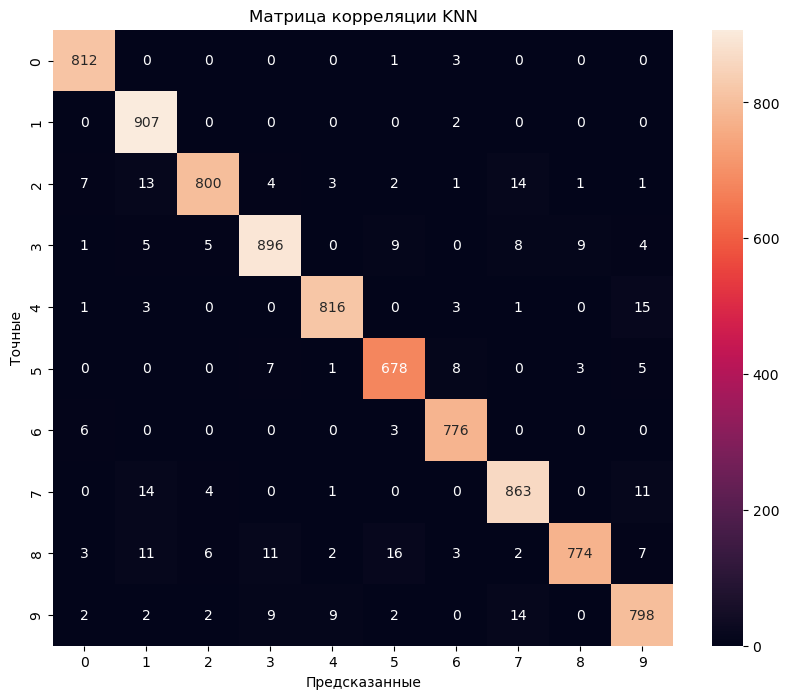
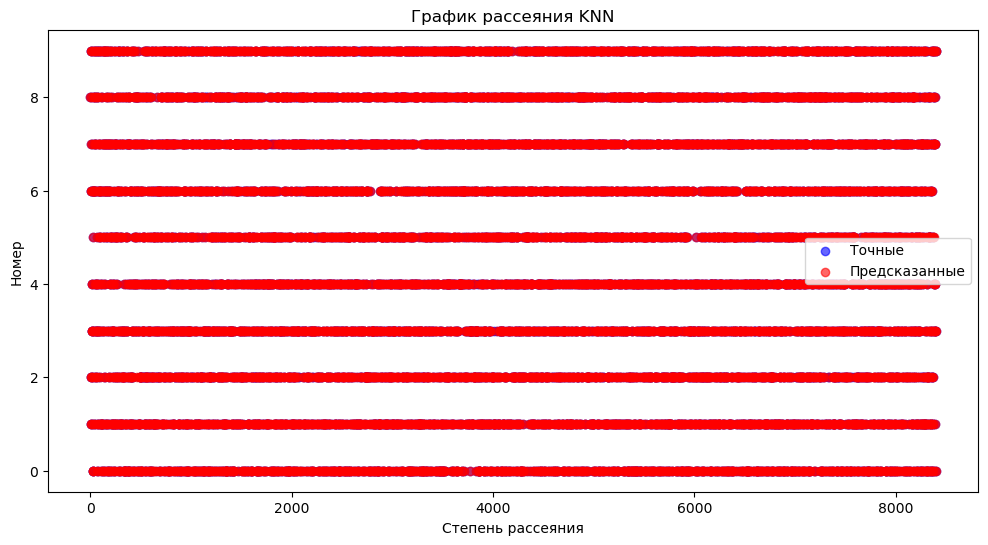


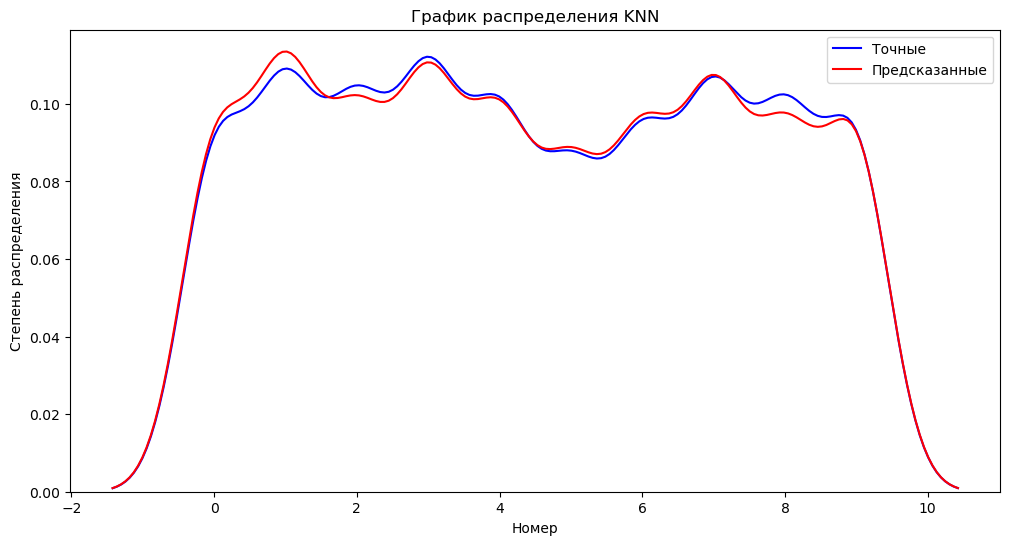
Рисунок 2.1.4. Матрица корреляции knn.

По матрице корреляции видно, что разброс минимален и составляет 4%. Следовательно, можно предположить, что метод К-ближайших соседей отличный метод для сортировки изображений в задачах классификации.

Приложим график рассеяния на рисунке 2.1.5. По нему видно, что предсказанные значения почти всегда находятся в одном месте с точными.

Рисунок 2.1.5. График рассеяния KNN.

Теперь график распределения по рисунку 2.1.6. По нему видно, что существуют проблемы с числовыми объектами: «1», «2», «8». В остальных случаях с остальными объектами всё впорядке.

Рисунок 2.1.6. График распределения KNN.

Рассмотрим гистограмму на рисунке 2.1.7. По ней видно больше. Числовые объекты «8», «2», «3», «4» содержат ошибки при определении. А числовые объекты «0», «1», «6», «7» по результатам сортировки изображений по категориям превышают своё количество.

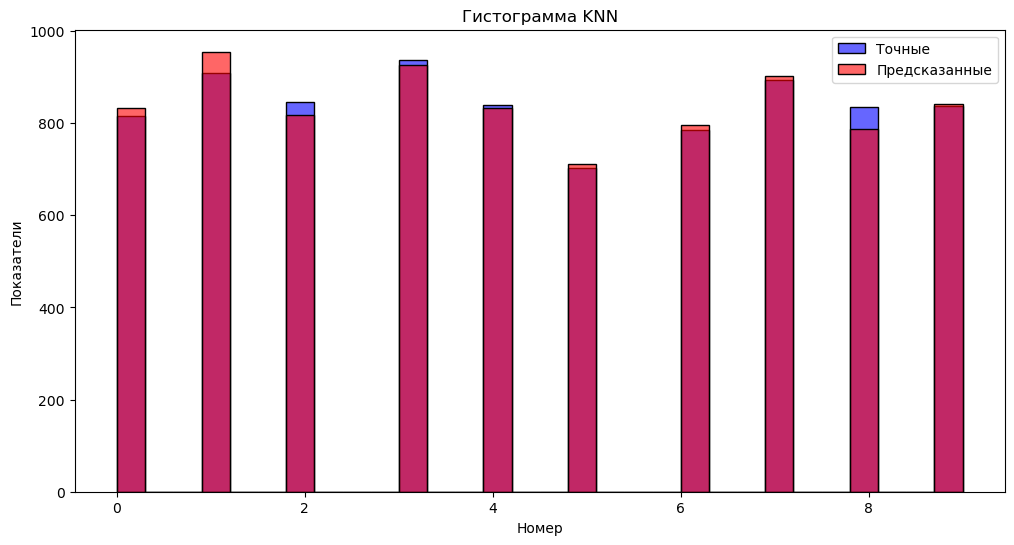


Рисунок 2.1.7. Гистограмма KNN.

## **Метод** **С-опорных векторов**

На Рисунке 2.2.1 изображены изображения после сортировки

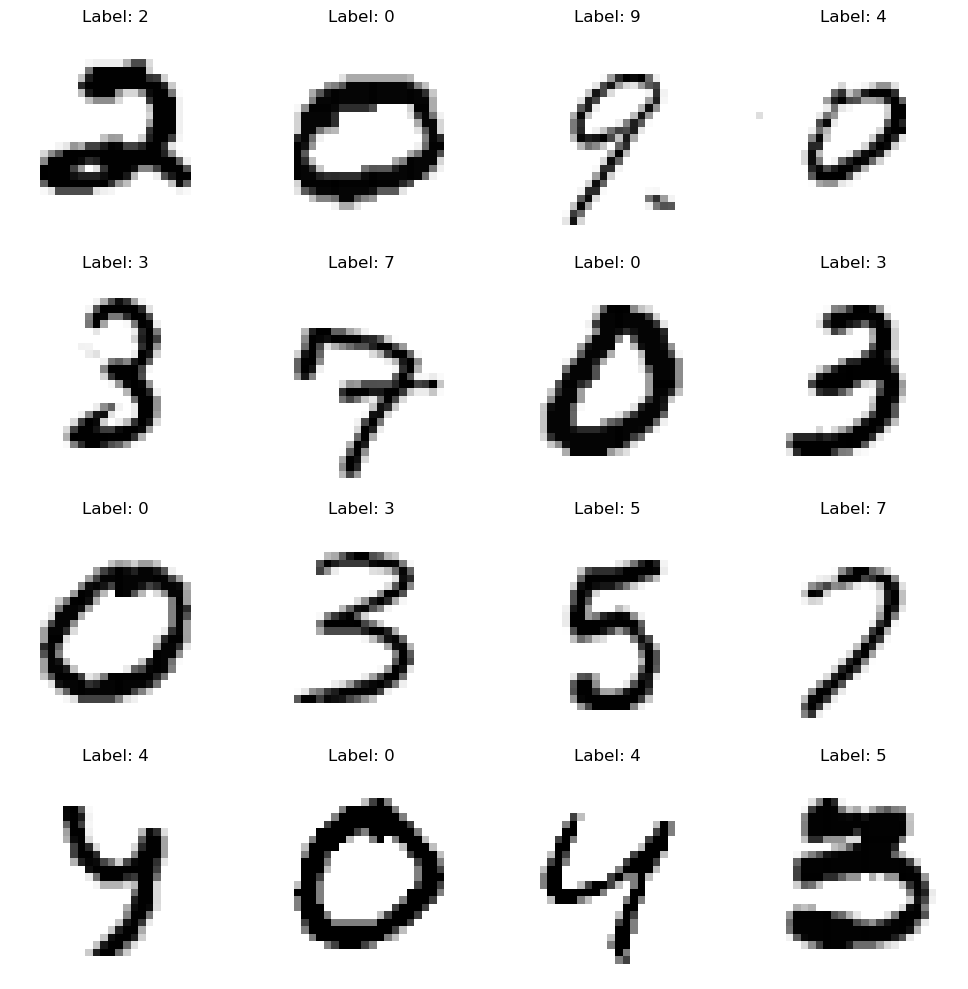


Рисунок 2.2.1 Рекогносцирование информации.

На первый взгляд оценка является довольно высокий, так как все изображения соответствуют своей категории.

Оценка точности данного метода состовляет 93%, как показано на рисунке 2.2.2. и является большой.

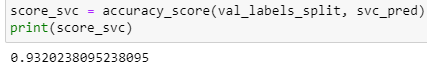


Рисунок 2.2.2. Оценка метода С-опорных векторов.

На рисунке 2.2.3. видно, что точность обученной модели высокая, но есть маленькие выбросы.

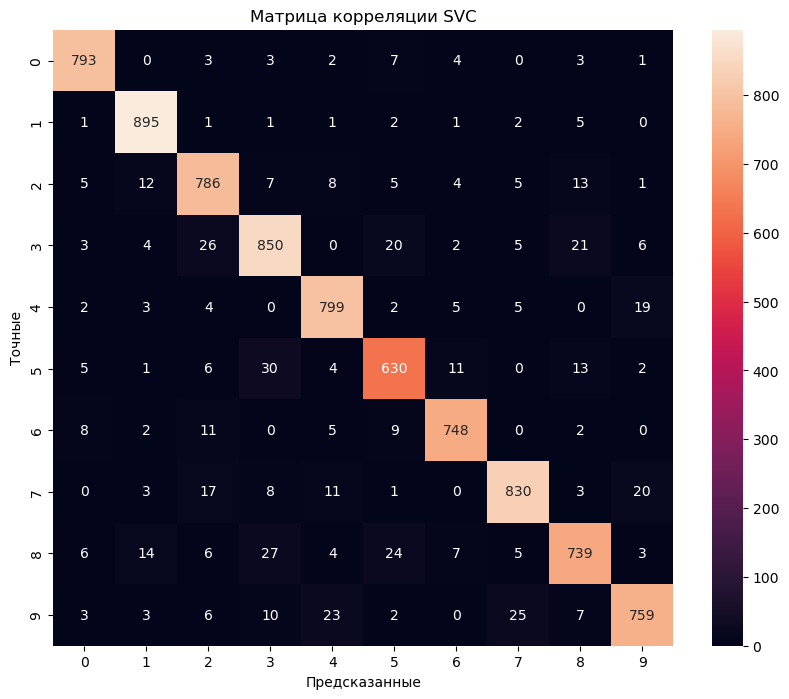


Рисунок 2.2.3. Матрица корреляции С-опорных векторов.

На рисунке 2.2.4. изображен график рассеяния и по нему видно, что точность модели большая. Я не вижу неточностей в предсказании.

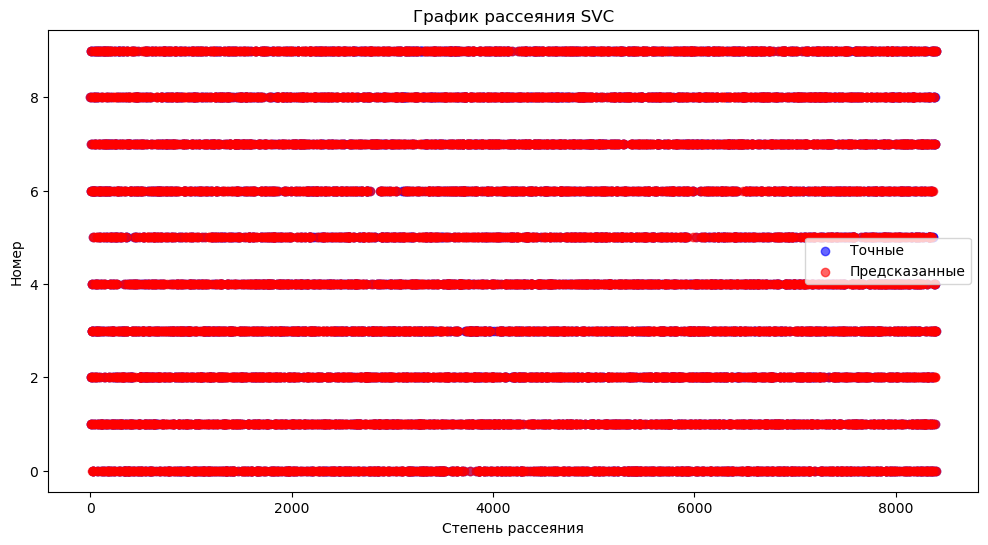


Рисунок 2.2.4 График рассеяния SVC

На рисунке 2.2.5. изображен график распределения модели метода С—опорных векторов. По результатам сортировки изображений моделью видно, что числовые объекты «1», «2», «4» превышают максимальное количество категории. А числовые объекты «7», «8», «9» не хватает до максимального количества.

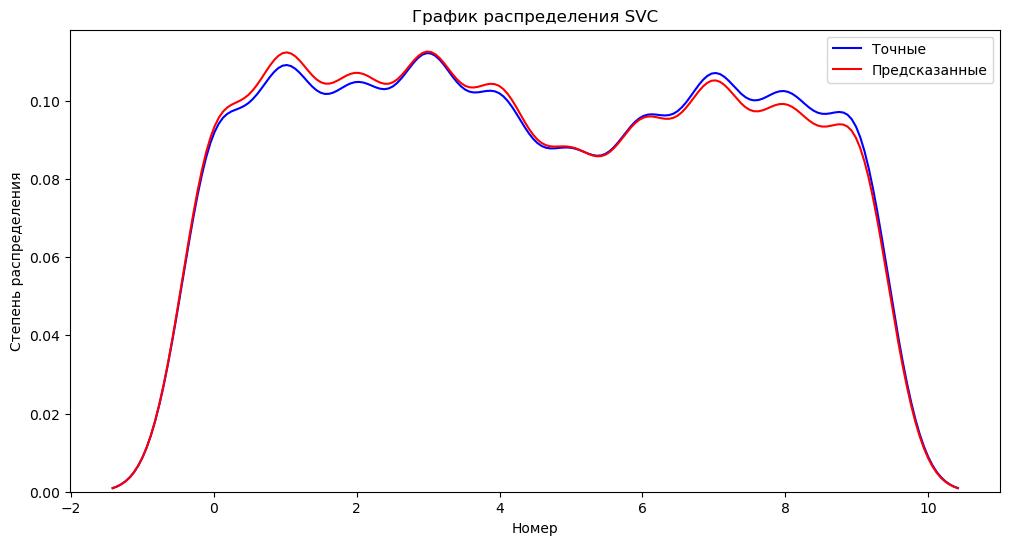


Рисунок 2.2.5. График распределения SVC

Переходим к гистограмме на рисунке 2.2.6.

По гистограмме видно превышения максимального количества у числовых объектов «0», «1», «2», «4». Точного количества числовых объектов «3», «5», «6». Нехватки «7», «8», «9».

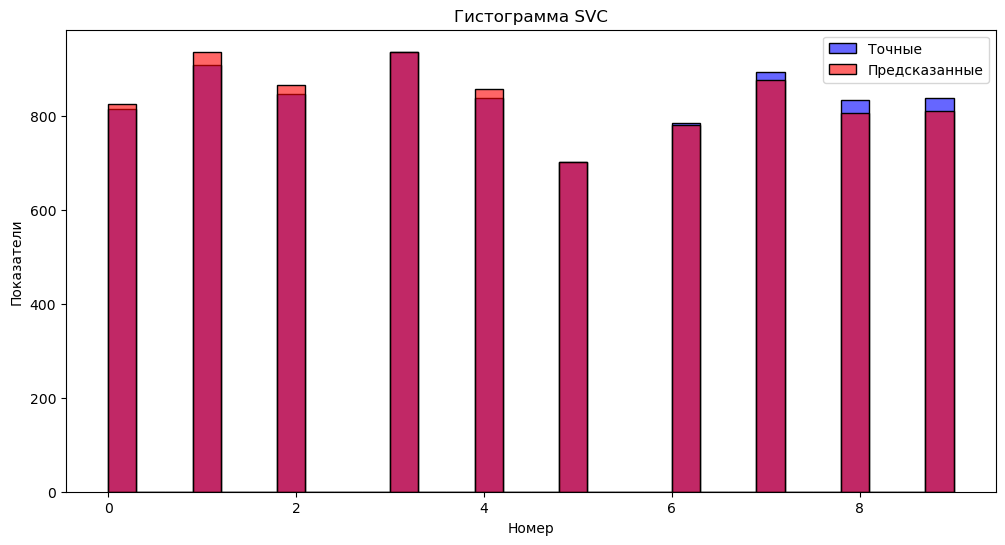
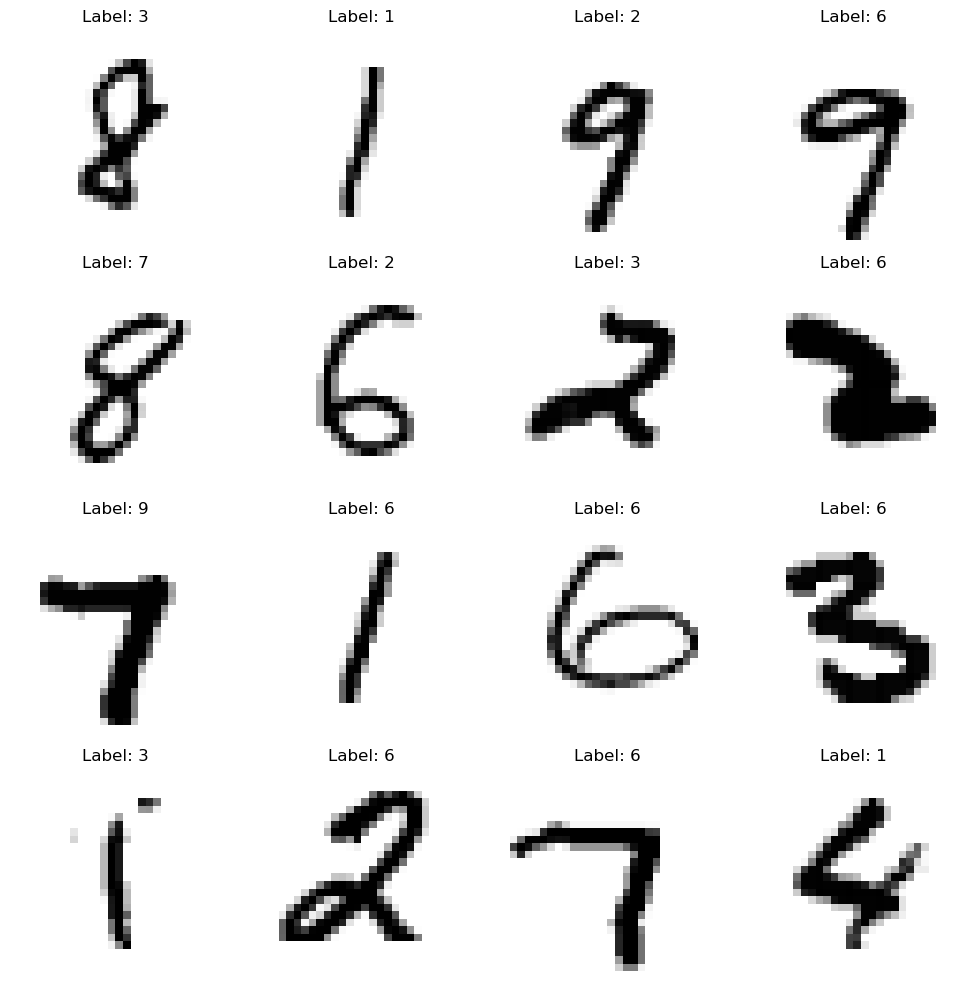


Рисунок 2.2.6. Гистограмма SVC.

## **Метод К-средних**

Алгоритм KMeans группирует данные и разделяет их на выборки на количество кластеров, у которых одинаковая дисперсия, тем самым минимизирует сумму квадратов внутри кластера.

На рисунке 2.3.1. видно, что модель опознала некоторые числовые объекты некорректно. В связи с этим следует ожидать среднюю оценку точности.

Рисунок 2.3.1. Изображение цифр К-средних.

На рисунке 2.3.2. Оценка точности модели К-средних составляет 59%. Ожидаемый непрактичный результат.

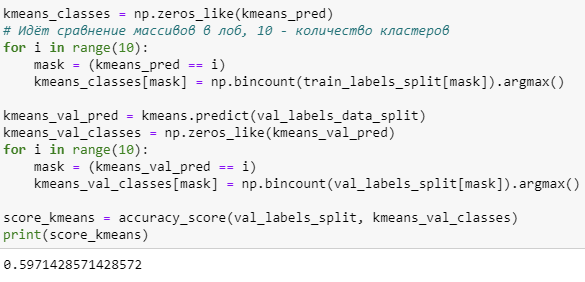


Рисунок 2.3.2. Оценка точности модели К-средних

По матрице корреляции высокая точность у числовых объектов «0», «1».

Отсутствуют совпадения полностью у числовых объектов «5» и «9». Средние показатели у «3», «4», «6», «7», «8». Выбросы большие. Предсказанные числовые объекты «3», «4» и «7» имеют большой разброс.

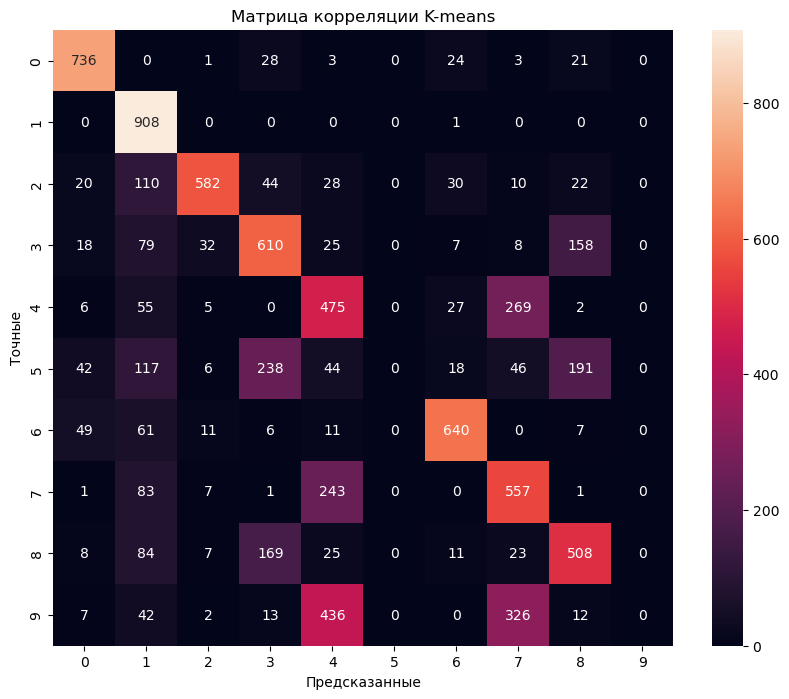


Рисунок 2.3.3 Матрица корреляции К-средних.

По рисунку 2.3.4. числовые объекты «5» и «9» не пересекаются с предсказанными значениями изображений.

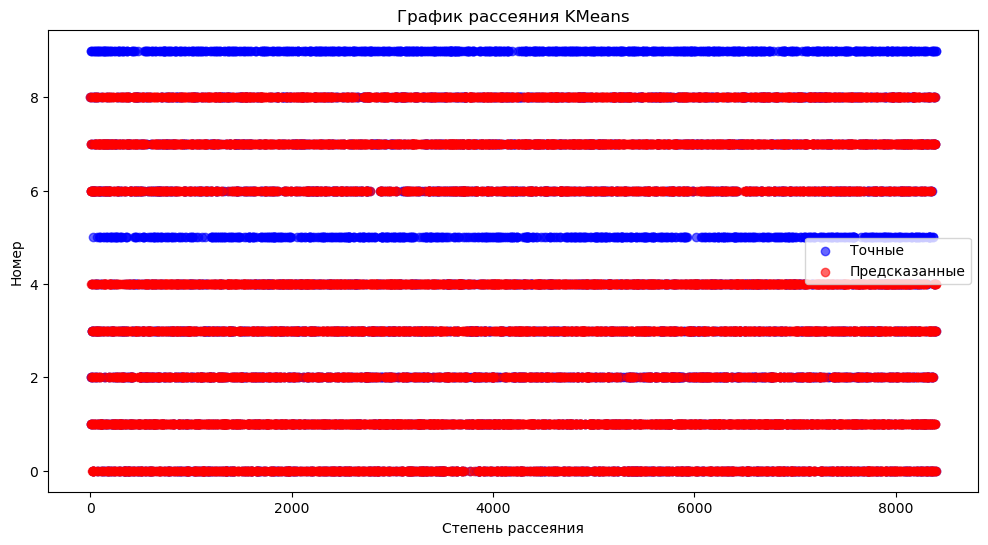


Рисунок 2.3.4. График рассеяния К-средних.

На рисунке 2.3.5. по графику распределения предсказанные числовые объекты меньше у «2», «5», «9». Предсказанное значение «5» минимально. По предсказанным числовым объектам «0», «1», «3», «4», «7», «8» превышают количество точных значений.

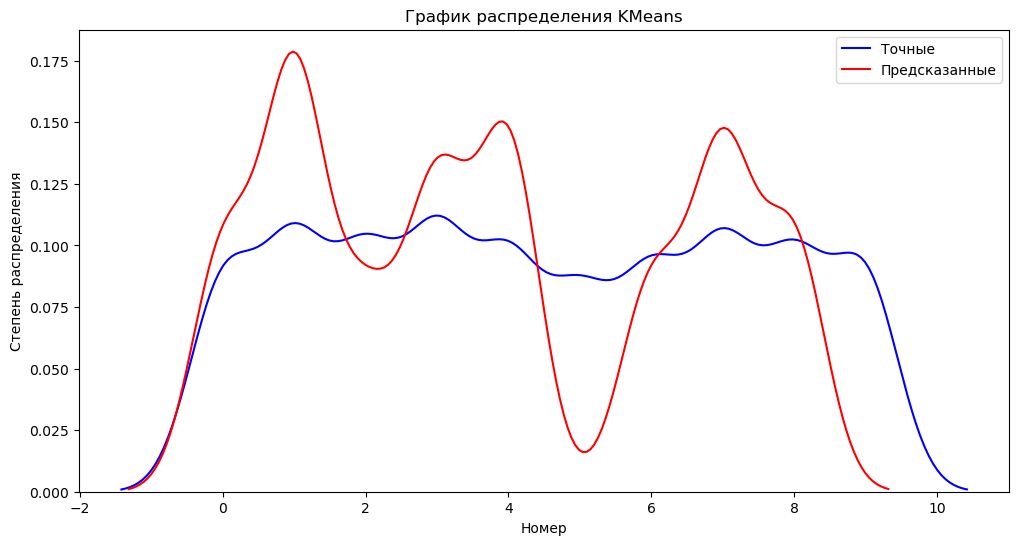


Рисунок 2.3.5. График распределения К-средних.

На рисунке 2.3.6. по гистограмме К-средних видно, что количество предсказанных значений превышает у «0», «3», «8».Сильно превышает «1», «4», «7».Пониженное количество предсказанных значений у «2», «6». Отсутствие предсказанных значений у «5» и «9».

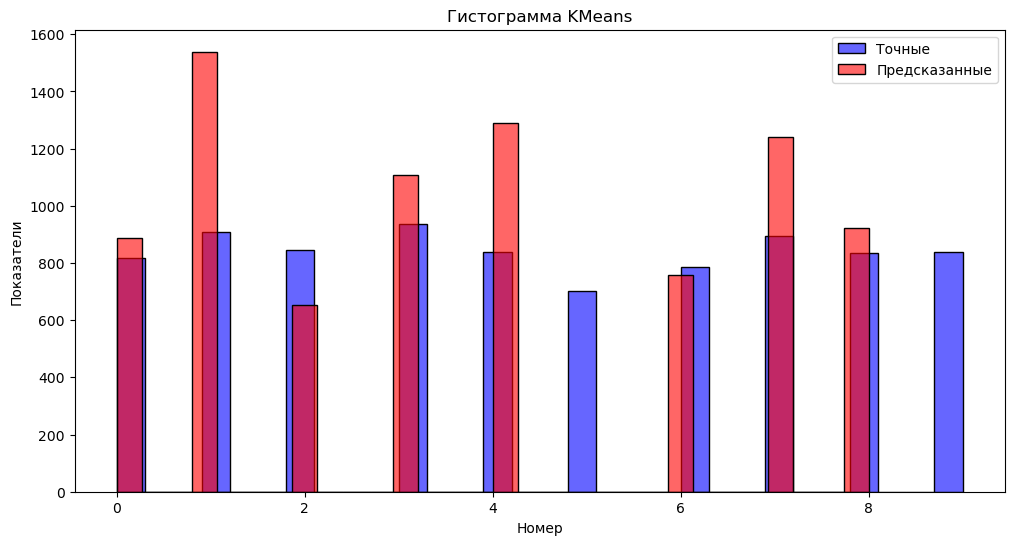


Рисунок 2.3.6. Гистограмма KMeans.

## **Метод Гауссовского наивного байесовского алгоритма**

На рисунке 2.4.1. видно, что модель относит изображения к категориям неверно в половине случаев. Отсюда исходит, что ожидаемая оценка точности модели будет низкой.

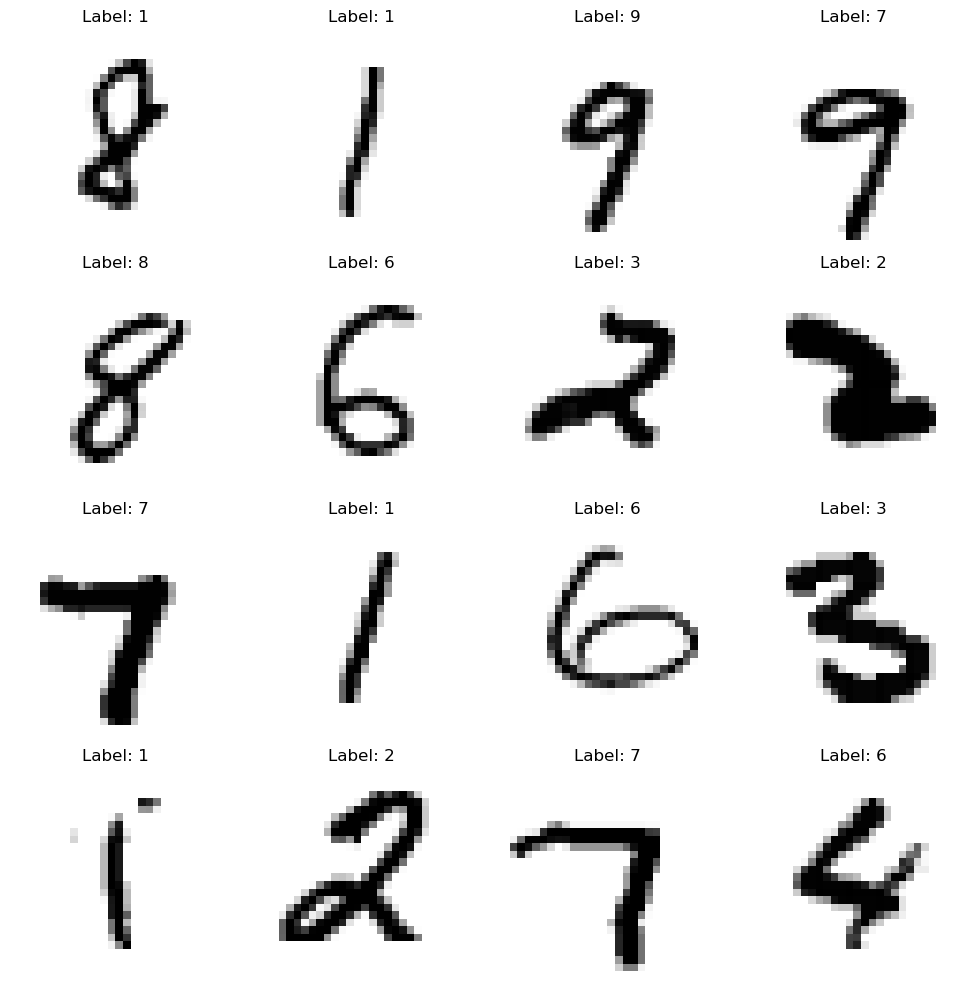


Рисунок 2.4.1. Изображение результата обработки методом Гауссовского наивного Байеса.

Из рисунка 2.4.2. мы видим оценку в 56% точности. Ожидаемо.

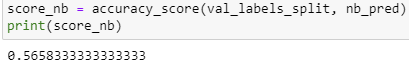
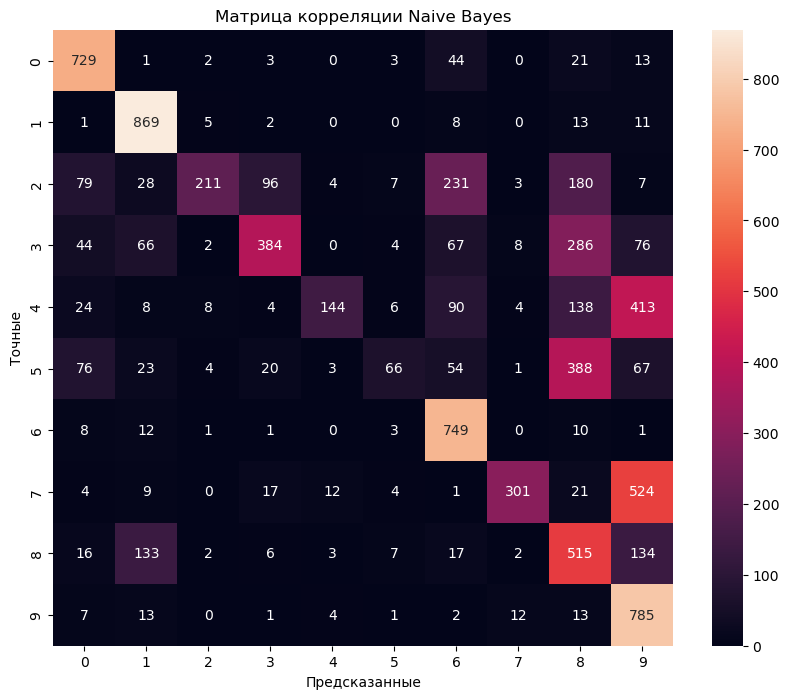


Рисунок 2.4.2. Оценка точности модели Гауссовского наивного Байеса.

На рисунке 2.4.3. мы видим матрицу корреляции модели данного метода. Имеется большой разброс у предсказанных чисел «6», «8», «9». Предсказанное число «5» имеет маленькую точность, путает с «8».

Рисунок 2.4.3. Матрица корреляции наивного Байеса.

На рисунке 2.4.4. видно, что не предсказанные данные есть у номеров: «2», «4», «7» и в большей степени «5». А также «3».

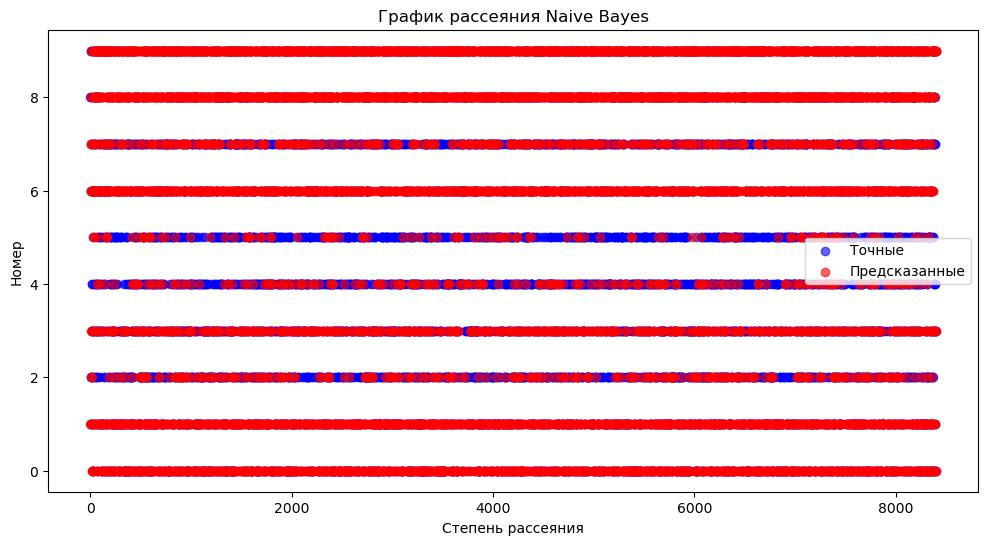


Рисунок 2.4.4. График рассеяния Наивного Байеса.

На рисунке 2.4.5. заметно, что предсказанные значения «0», «6», «8», «9» превышают количество точных значений. А также предсказанные значения «2», «3», «4», «5», «7» меньше максимального количества правильных ответов.

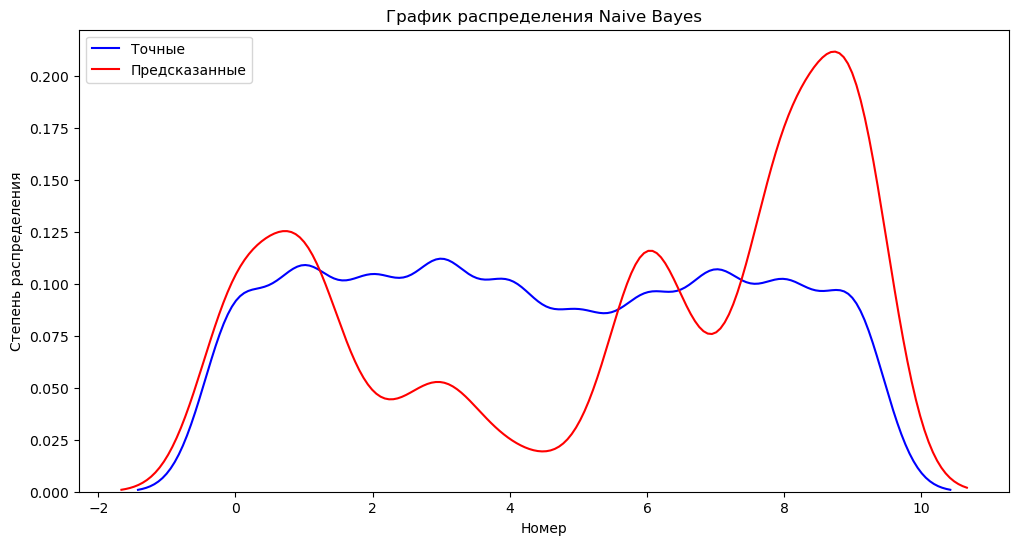


Рисунок 2.4.5. График распределения наивного Байеса.

По рисунку 2.4.6. можно сказать, что распознанные категории по числовым объектам «2», «3», «4», «5», «7» меньше примерно в половину количества правильных ответов. А по числовым объектам «0», «1», «6», «8», «9» превышают, на цифрах «8» и «9» превышает в 2 раза.

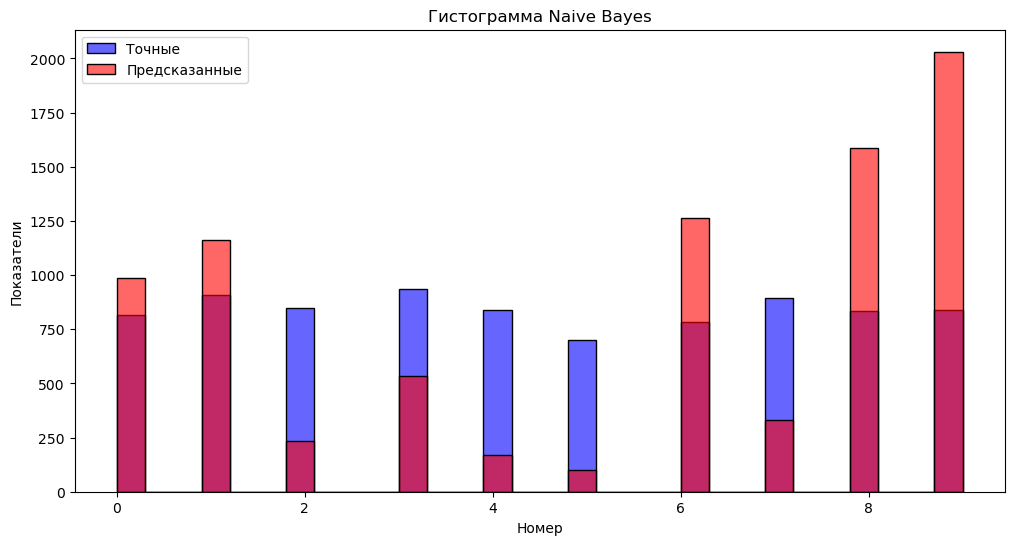


Рисунок 2.4.6. Гистограмма наивного байеса.

## **Метод случайного леса**

На рисунке 2.5.1. присутствует 1 неверно определенный элемент, предполагаемая оценка точности модели метода случайного леса высокая.

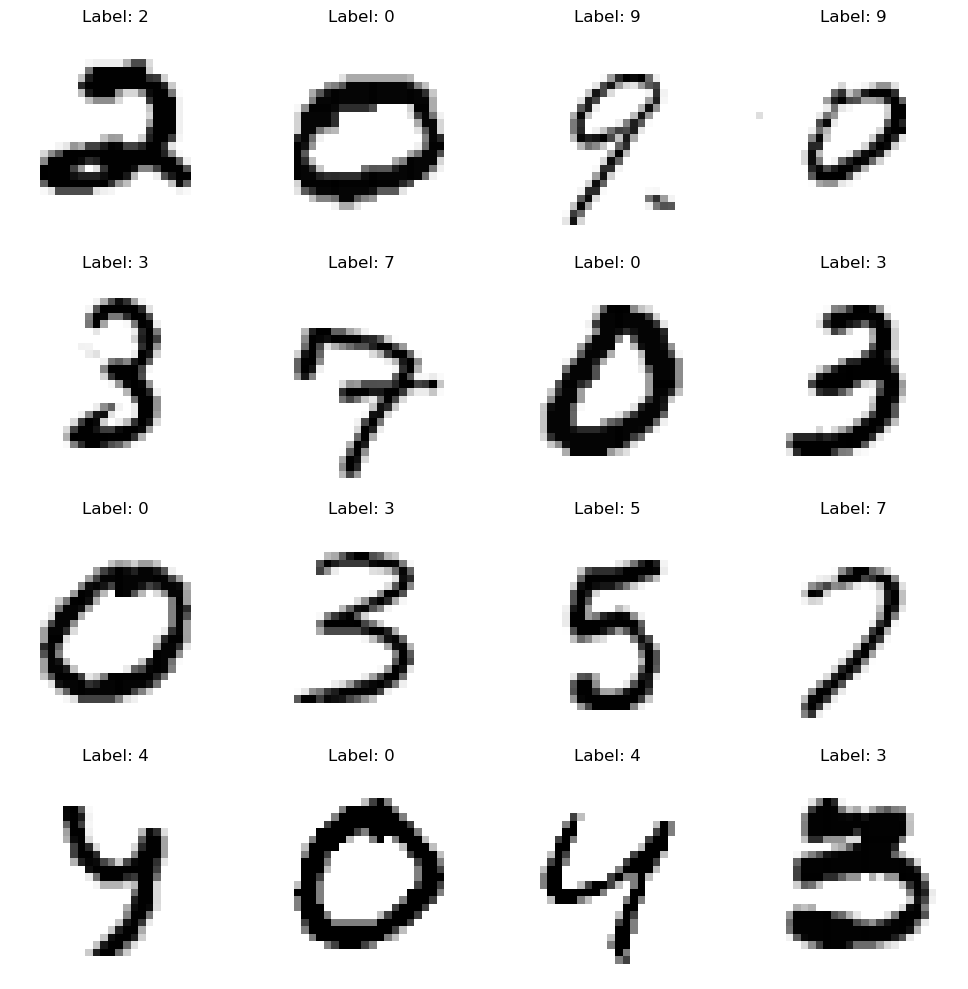


Рисунок 2.5.1. Изображение, полученное от модели случайного леса.

На рисунке 2.5.2. Мы видим высокую оценку в 93% точности.

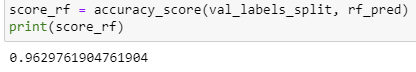


Рисунок 2.5.2. Оценка метода случайного леса.

По матрице корреляции на рисунке 2.5.3. видно, что разброс минимален большинство данных распознано верно.

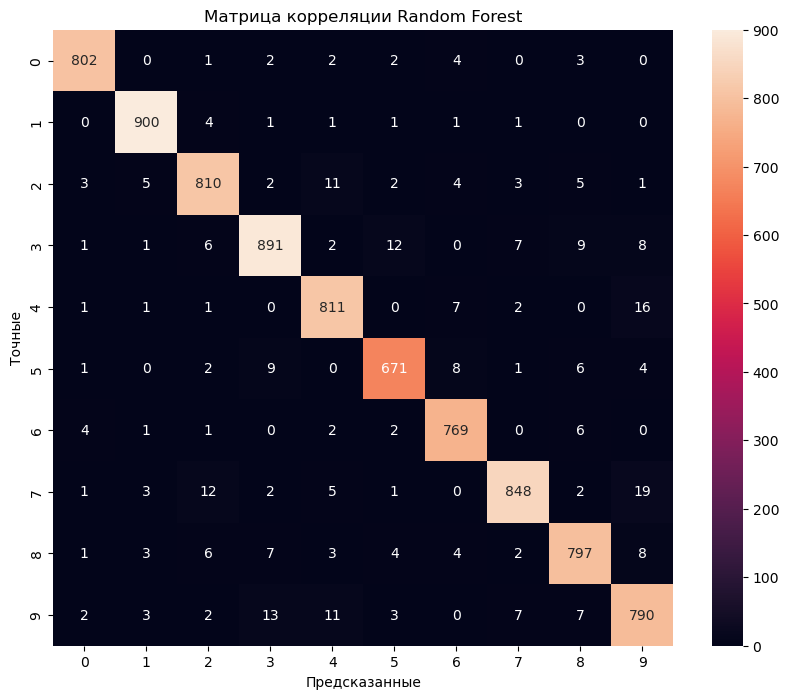


Рисунок 2.5.3. Матрица корреляции случайного леса.

По графику рассеяния случайного леса на рисунке 2.5.4. не видно неверно указанных предсказаний.

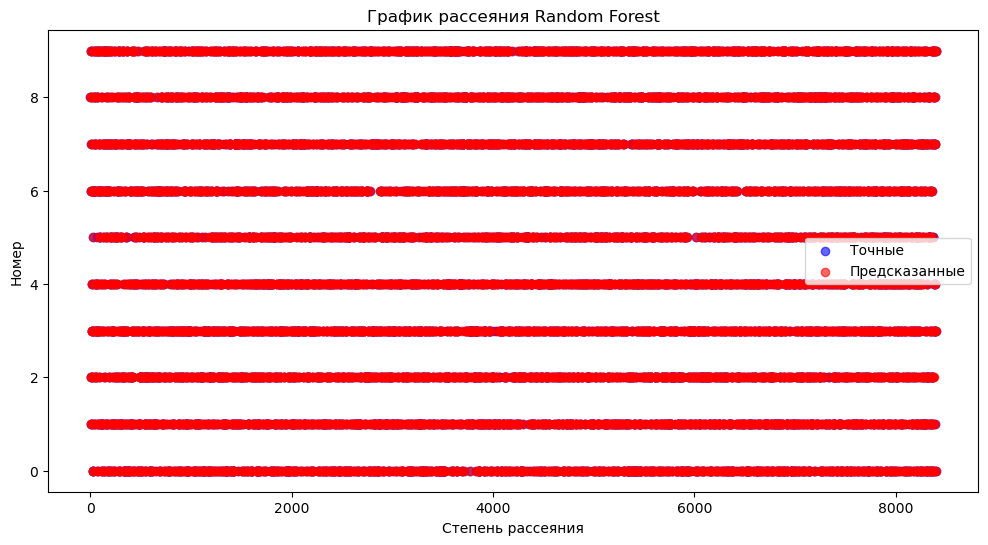


Рисунок 2.5.4. График рассеяния случайного леса.

На рисунке 2.5.5. видно, что цифра «7» ненамного превышает количество максимального количество точных «7».

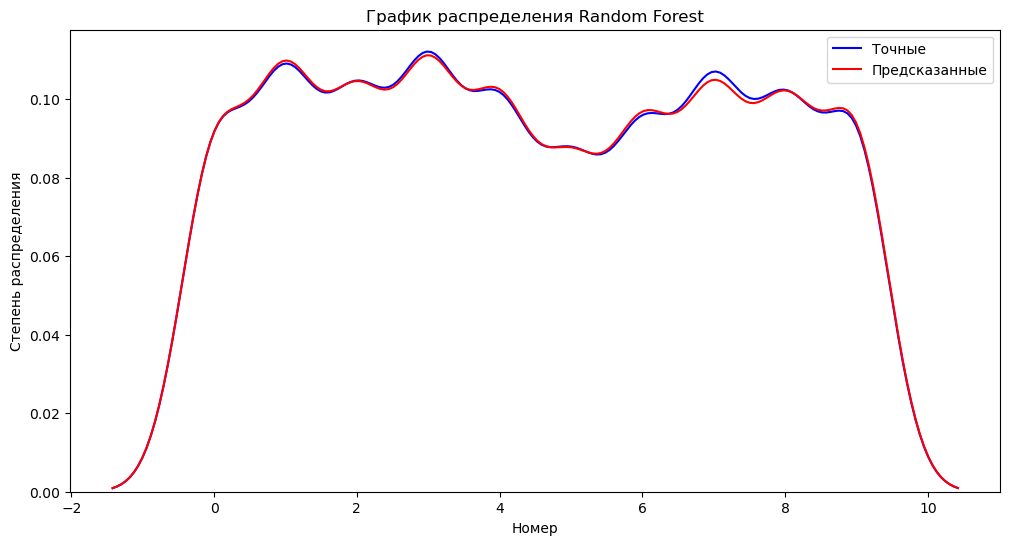


Рисунок 2.5.5. График распределения случайного леса.

На рисунке 2.5.6. видно легкое превышение предсказанных значений у цифр «2», «4», «6», «9». И нехватка у цифр «3», «7».

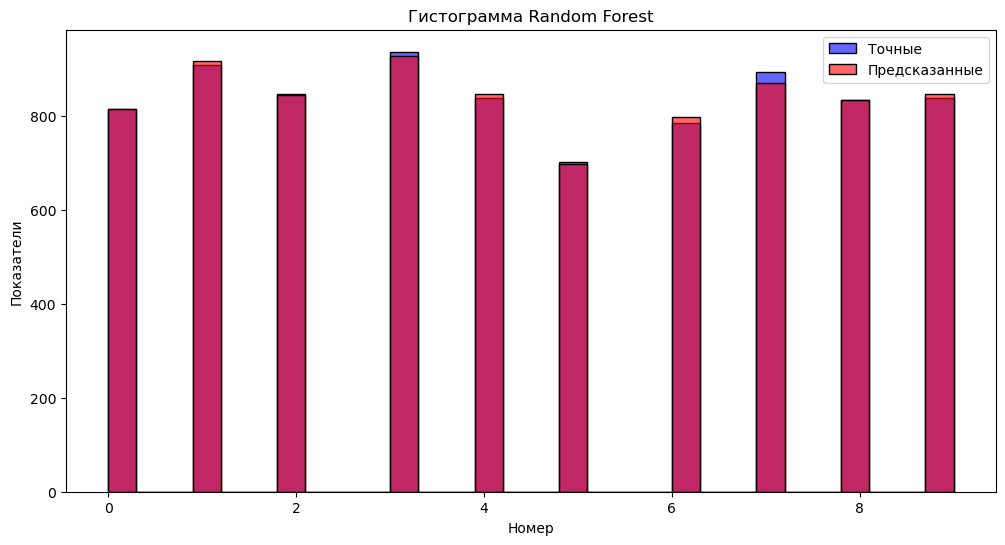


Рисунок 2.5.6. Гистограмма случайного леса

## **Метод С-опорных векторов линейного ядра**

На рисунке 2.6.1 видно много несоответствий с правильными числами, ожидаемая оценка средняя, либо низкая.



Рисунок 2.6.1. Изображение результата С-опорных векторов линейного ядра.

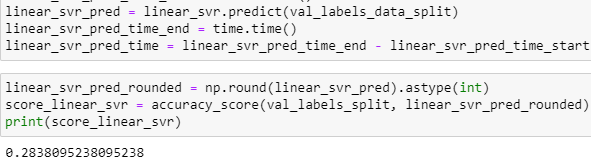
На рисунке 2.6.2. изображена низкая оценка в 28%.

Рисунок 2.6.2. Оценка точности метода LinearSVR.

На рисунке 2.6.3. видно, что цифра «3» и «5» больше других распознались правильно. Разброс большой вокруг правильных вариантов. Также присутствуют несуществующие значения. Не были распознаны «0», «1». Скорее всего это связано с тем, что это метод кластеризации.

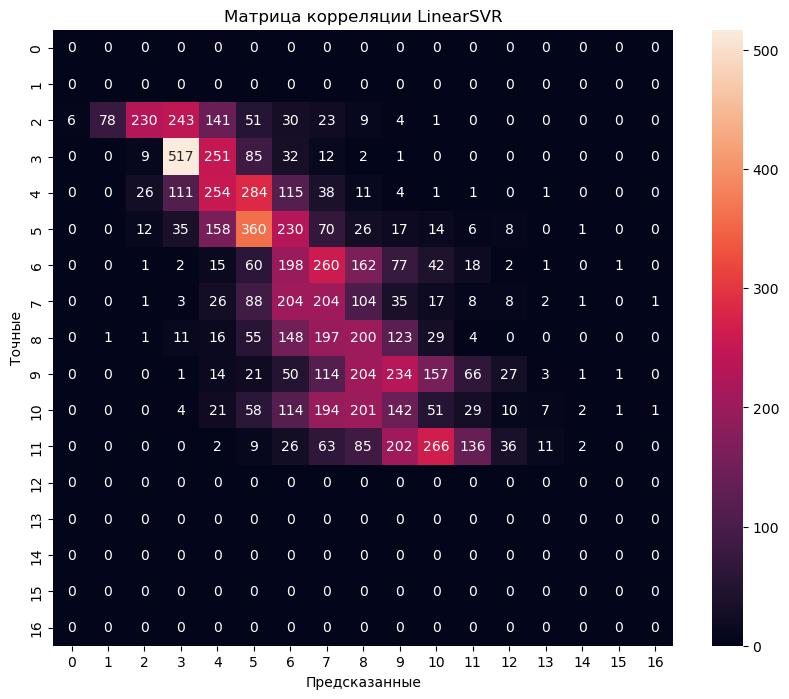


Рисунок 2.6.3. Матрица корреляции С-опорных векторов с линейным ядром.

На рисунке 2.6.4. такая же история, есть ошибки у цифр «0», «9».

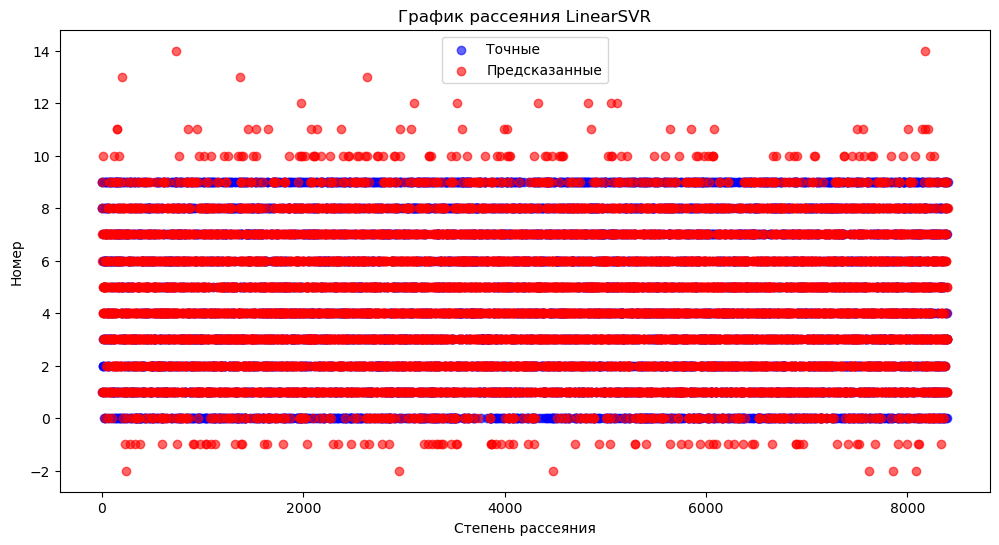


Рисунок 2.6.4. График рассеяния С-опорных векторов с линейным ядром.

На рисунке 2.6.5. Цифры от «7» до «10» меньше количеством правильного варианта. Цифры от «1» до «7» превышают максимальное количество. Цифра «0» почти отсутствует.

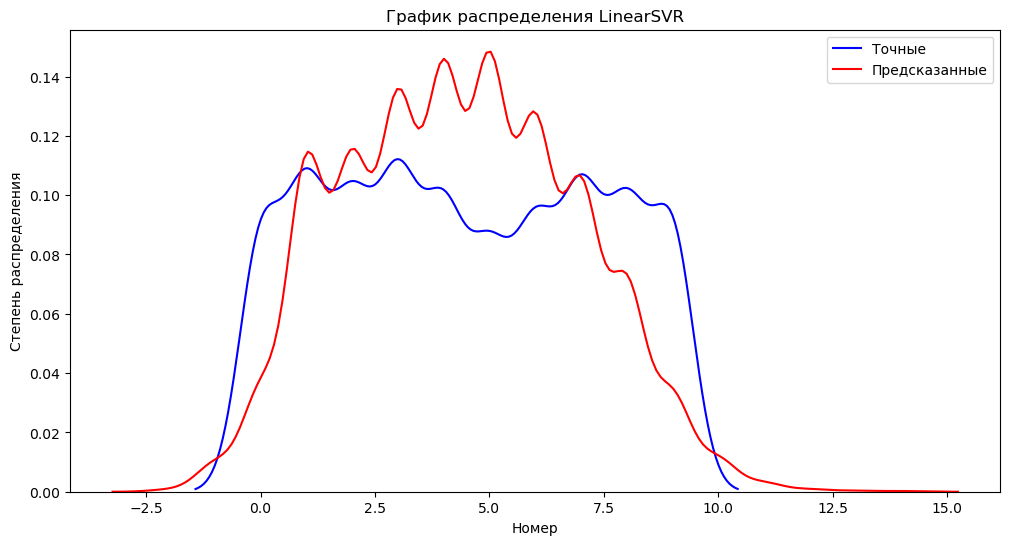


Рисунок 2.6.5. График распределения С-опорных векторов с линейным ядром.

На рисунке 2.6.6. видно, предсказанные числа превышают у «1», «2», «3», «4», сильно у «5», превышает также у «6». Цифры «0» и «9» сильно отстают от правильного распознавания. Не так сильно отстают «8» и «7».

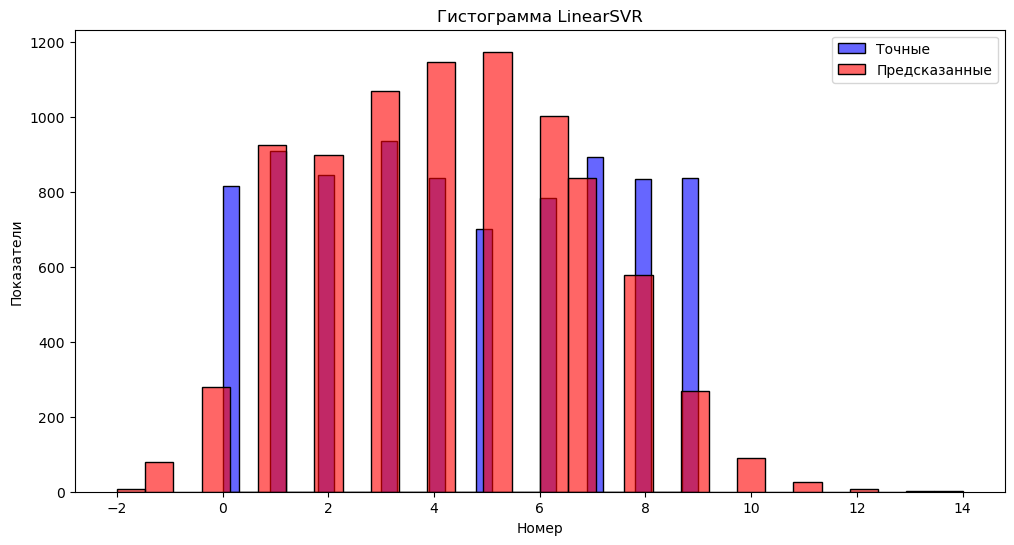


Рисунок 2.6.6. Гистограмма С-опорных векторов с линейным ядром.

## Общая оценка моделей

Мы в коде засекли время начала и окончания обучения всех моделей. По графику времени самое долгое был метод С-опорных векторов, за ним шёл метод случайного леса и методы К-средних и С-опорных линейного ядра одинаковы. Самый быстрое обучение произошло у метода ближайших соседей, после неё идёт метод Гауссовского наивного Байеса [Рисунок 2.7.1.].

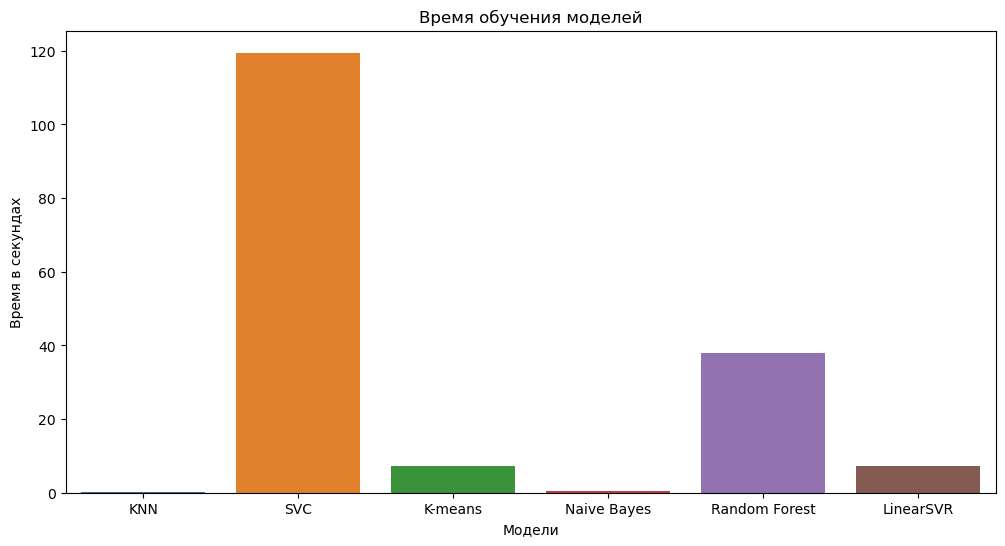


Рисунок 2.7.1. Время обучения моделей.

Далее, перейдём к графику оценки времени сортировки изображений по категориям на рисунке 2.7.2. Самым долгим был метод С-опорных векторов и занимал почти 120 секунд. По быстроте обработки лидирует метод ближайших соседей, после него идёт метод С-опорных векторов с линейным ядром. Далее по порядку: метод К-средних, метод случайного леса, метод Гауссовского Наивного Байеса.

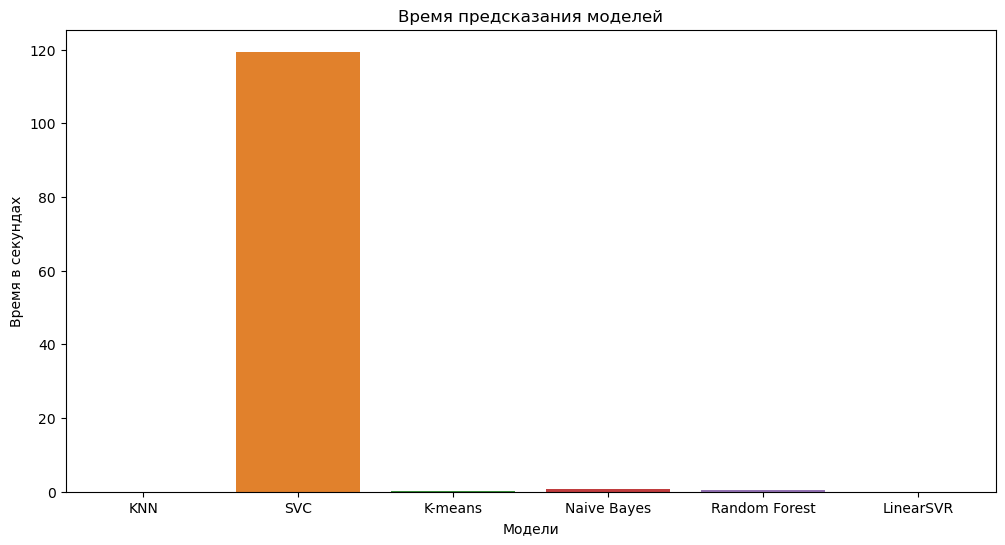


Рисунок 2.7.2. Время предсказания моделей.

На рисунке 2.7.3. график сравнения оценки обученных моделей на соответствие. Первое место занимает метод ближайших соседей с 96% точности. Второе место занимает метод случайного леса с 93%, далее самый долгий метод С-опорных векторов.

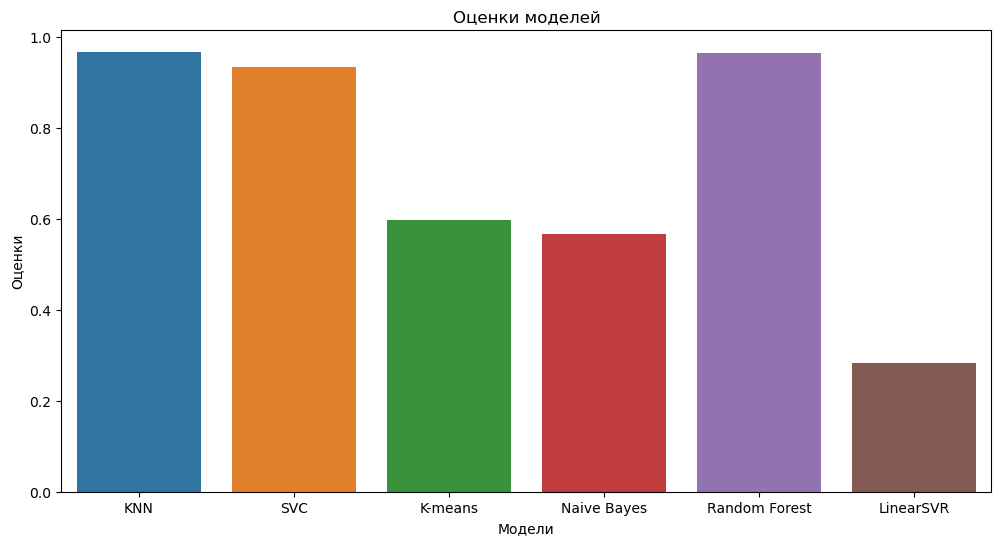


Рисунок 2.7.3. Сравнение оценки моделей.

# ЗАКЛЮЧЕНИЕ

В данной курсовой работе были достигнуты следующие цели :

* Обучено 6 различных моделей различными методами;
* Был рассмотрен и проанализирован метод кластеризации в задачах классификации;
* Были использован метод машинного обучения без учителя.

В данной программе были задействованы такие библиотеки, как :

* “sklearn”;
* “seaborn”;
* “Matplotlib”.

В разработке программы с машинным обучением для распознавания объектов на изображениях в задачах классификации были реализованы следующие вопросы и функционал:

- Машинное обучение методов распознавания;

- Количественный и качественный анализ каждой модели;

- Общий сравнительный анализ методов модели;

- Программа реализована на архитектуре строения однослойной нейронной сети без глубокого обучения;

- Программа поддерживает запуститься на любом устройстве, где установлен Python 3.5 и выше, кроме ARMv6, Windows ARM64.

Поставленные цели были достигнуты, программа позволяет пользователю:

Произвести анализ методов моделей машинного обучения.  
 Код в репозитории github по ссылке : https://github.com/neinzwein/FinUni/tree/main/4/

# **Источники**

Основная литература

* [М.В. Коротеев. Об основных задачах дескриптивного анализа данных.](https://drive.google.com/file/d/1JoHxg3dfc53bSZtsNfPPB5Es_C4Q9hJ0/view?usp=sharing)
* [М.В. Коротеев. Учебное пособие по дисциплине “Анализ данных и машинное обучение” - 2018.](https://portal.fa.ru/Files/Data/85bca4b4-cd40-4dae-bb62-dbd6c04fa0a0/um_tehnanalizadannihimashobuchenie_18.pdf)

Дополнительная литература

* A. Geron. Hand on Machine Learning with scikit-learn and Tensorflow - 2017 (564p)
* C. Albon. Machine learning with Python Handbook - 2018 (427p)
* L.P. Coelho, W. Richert. Building machine learning systems with Python - 2015 (326p)
* J. Grus. Data science from scratch - 2015 (330p)
* W. McKiney. Pandas: powerful Python data analysis toolkit - 2016 (1971p)

Видео-ресурсы

* [Классический курс по машинному обучению](https://www.youtube.com/playlist?list=PLLssT5z_DsK-h9vYZkQkYNWcItqhlRJLN)
* [Конспекты](http://cs229.stanford.edu/syllabus.html)
* [Материалы на гитхабе](https://github.com/vkosuri/CourseraMachineLearning)
* [Pandas tutorial](https://www.youtube.com/playlist?list=PLeo1K3hjS3uuASpe-1LjfG5f14Bnozjwy)
* Канал [Python programmer](https://www.youtube.com/user/consumerchampion/playlists) - много контента по программированию на питоне
* Серия [Data analysis](https://www.youtube.com/playlist?list=PLzH6n4zXuckpfMu_4Ff8E7Z1behQks5ba) от Computerphile
* https://youtu.be/8mkh4uGxNfo?si=5cyURy8e9DlOi065

Интернет-ресурсы

* <https://www.kaggle.com/competitions/digit-recognizer>