

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных систем

Факультет автоматики и вычислительной техники

Кафедра систем автоматизации управления

Классификация данных набора MNIST

Отчет по лабораторной работе №2

по дисциплине

«Теория информации, данные, знания»

Выполнил студент гр. ИТб-4302-02-00 _____ /Вершинин П.А./
(Подпись)

Руководитель к.т.н., доцент _____ /Чистяков Г.А./
(Подпись)

Работа защищена с оценкой «_____» «___» _____ 2023 г.

1 Описание лабораторной работы

Цель: получение студентами навыков работы с классификацией данных набора MNIST машинного обучения библиотекой sklearn на языке программирования Python.

В ходе работы необходимо выполнить следующее задание:

- импорт библиотек и загрузка данных;
- подготовка данных;
- выбор модели;
- обучение модели;
- предсказания и создание файла для оценки результата;

2 Задание

Каждое изображение имеет 28 пикселей в высоту и 28 пикселей в ширину, что составляет в общей сложности 784 пикселя. С каждым пикселем связано одно значение пикселя, указывающее на светлоту или затемненность этого пикселя, причем более высокие цифры означают более темный цвет. Это значение в пикселях представляет собой целое число в диапазоне от 0 до 255 включительно.

Набор обучающих данных (train.csv) содержит 785 столбцов. Первый столбец, называемый "метка", — это цифра, которая была нарисована пользователем. Остальные столбцы содержат значения в пикселях соответствующего изображения.

Прогноз должен представлять собой: строку содержать идентификатор изображения и цифру, которую прогнозируем.

Формат выглядит следующим образом на рисунке 1

```
ImageId,Label
1,3
2,7
3,8
(27997 more lines)
```

Рисунок 1 – Формат данных в конечном виде

3 Импорт библиотек и загрузка данных

Для начала работы необходимо импортировать следующие библиотеки: «Pandas», «scikit-learn» (sklearn). На рисунке 2 отображен импорт данных библиотек.

«SVC» – это метод машинного обучения из библиотеки «scikit-learn» (sklearn), который используется для задач классификации. Основан на алгоритме опорных векторов «SVM» и работает путем нахождения оптимальной гиперплоскости, которая максимально разделяет два класса данных.

```
import pandas as pd
from sklearn.svm import SVC
```

Рисунок 2 – Импорт библиотек

Загрузка данных из файлов «train.csv» и «test.csv» происходит как показано на рисунке 3.

```
data_train = pd.read_csv('train.csv')
data_test = pd.read_csv('test.csv')
```

Рисунок 3 – Загрузка данных из файлов

«data_train» содержит обучающие данные, представленные в виде матрицы признаков.

«data_test» содержит данные, для которых нужно выполнить предсказания.

4 Подготовка данных

Перед анализом данных необходимо разделить обучающие данные на входные признаки и целевые значения или категории. В данном случае, из обучающей выборки целевыми значения будут в столбце «label», и входные признаки в остальных столбцах. Пример разделения данных представлен на рисунке 4.

```
data_train_X = data_train.drop(columns='label')  
data_train_Y = data_train['label']
```

Рисунок 4 – Разделение обучающего набора данных

Обучающий набор данных обладает значительным размером, что обеспечивает модели достаточное количество примеров для обучения. Это позволяет модели рассмотреть больше разнообразных ситуаций и обучиться более надежно.

Нормализация признаков может быть полезной в некоторых случаях для улучшения сходимости алгоритмов машинного обучения. Однако, в некоторых случаях она может не приносить выгоды, как это случилось в данной работе. На рисунке 5 представлены результаты оценки обученных моделей на тестовых данных.




	submission.csv Complete · now · с нормализацией StandardScaler()	0.65939
	submission.csv Complete · 20m ago · model = SVC(C=1.2, kernel='poly', gamma='au...	0.97775
	submission.csv Complete · 27m ago · model = SVC(random_state=25_555)	0.97521

Рисунок 5 – Результаты оценок моделей с и без использования нормализации

В первом «submission.csv» используется нормализация данных с помощью метода «StandardScaler()» из библиотеки «sklearn». «StandardScaler» – этот метод стандартизирует данные, приводя их к нулевому среднему и единичному стандартному отклонению. Оценка точности результата равна 0,65939.

В остальных «submission.csv» не используется нормализация данных. Оценка точности результата равна 0,97.

Таким образом, нормализация данных отрицательно влияет на оценку выходных результатов.

5 Выбор модели и обучение

Для решения задачи классификации, выбран классификатор Support Vector Classification «SVC» из библиотеки «scikit-learn». «SVC» может использовать различные типы ядер, такие как линейное, полиномиальное и радиальное базисное функциональное «RBF», чтобы обрабатывать как линейно, так и нелинейно разделимые данные. «SVC» позволяет настраивать параметры, такие как «C», чтобы управлять компромиссом между максимизацией зазора и минимизацией ошибки классификации.

✓	submission.csv Complete · 20m ago · model = SVC(C=1.2, kernel='poly', gamma='auto', degree=2, coef0=0.8, random_state=25...	0.97775
✓	submission.csv Complete · 27m ago · model = SVC(random_state=25_555)	0.97521
✓	submission.csv Complete · 15d ago · model = SVC(kernel='poly', gamma='scale', coef0=0.8, random_state=25_555)	0.97778
✓	submission.csv Complete · 15d ago · SVC(kernel='poly', gamma='scale', coef0=0.5, random_state=25_555)	0.97825
✓	submission.csv Complete · 15d ago · KNC n_neighbors = 5	0.967
✓	submission.csv Complete · 15d ago · KNC n_neighbors = 20	0.95828
✓	submission.csv Complete · 15d ago · KNN n_estimators = 10	0.96414

Рисунок 6 – Результаты оценок моделей с разными параметрами

Данный классификатор имеет наиболее высокую оценку результатов из следующих классификаторов, которые использовались: «KNeighborsClassifier».

Параметр «kernel» в методе опорных векторов «SVM» определяет тип функции ядра.

Изменив тип ядра «SVC» с линейного на полиномиальное результат улучшился. Полиномиальное ядро позволяет SVM модели создавать нелинейные разделяющие гиперплоскости в данных. Вместо того, чтобы искать прямую линию, как это делается с линейным ядром «kernel='linear'», полиномиальное ядро позволяет модели учитывать нелинейные зависимости между признаками.

Параметры, которые можно настроить для полиномиального ядра в «sklearn», включают следующие:

1. «degree»: Этот параметр определяет степень полинома, которая будет использоваться для преобразования данных. Например, если degree=2, то будет использоваться квадратичное полиномиальное преобразование данных.

2. «coef0»: Этот параметр определяет свободный член в полиноме. Он позволяет настроить смещение (свободный член) в функции ядра.

3. «gamma»: Этот параметр, контролирует форму функции ядра и влияет на то, насколько сильно модель будет приспосабливаться к данным.

Обучение модели выполняется с использованием обучающего набора данных. Используем следующий код для создания и обучения модели (рис. 6).

```
model = SVC(kernel='poly', gamma='scale', coef0=0.5, random_state=25_555)
model.fit(data_train_X, data_train_Y)
```

Рисунок 6 – Обучение модели

6 Предсказание и создание файла для оценки результата

После успешного обучения модели на обучающем наборе данных, можно перейти к предсказаниям классов на тестовых данных. В данной работе используем обученную модель «SVC» для этой цели (рис. 7).

```
id_values = list(range(1, data_test.shape[0] + 1))
ids = pd.DataFrame(columns=["ImageId", "Label"])
ids["ImageId"] = id_values
ids["Label"] = model.predict(data_test)
ids.to_csv(path_or_buf="submission.csv", index=False)
```

Рисунок 6 – Предсказание классов для тестового набора данных и запись в заданный формат

Данный код выполняет предсказания для всех объектов в тестовом наборе данных и сохраняет результаты в переменной «ids». Для оценки результатов и подготовки их к отправке или анализу, создается CSV-файл, который содержит идентификаторы объектов «ImageId» и предсказанные моделью метки классов «Label». В результате выполнения данного кода будет создан файл

"submission.csv", который содержит предсказания для каждого объекта в тестовом наборе данных. Этот файл можно использовать для отправки результатов задачи или для дальнейшего анализа. На рисунке 7 представлен результат анализа данных.

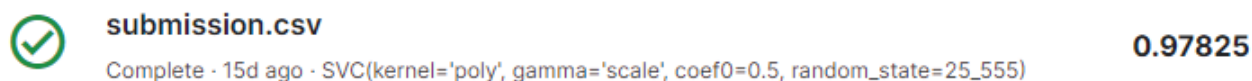


Рисунок 7 – Результат оценки обученной модели без использования нормализации данных

Критерием оценки для этого конкурса является точность категоризации, или доля тестовых изображений, которые были правильно классифицированы. Например, точность категоризации, равная 0,97, указывает на то, что вы правильно классифицировали все изображения, кроме 3%.

В итоге получаем оценку в 0,97825.

Вывод

В ходе выполнения работы были освоены основы работы с классификацией данных набора MNIST машинного обучения библиотекой «scikit-learn». Это включает в себя загрузку данных, предварительную обработку, выбор модели, обучение, оценку и предсказания. Важно также учитывать нормализацию данных и подбор параметров модели для достижения лучших результатов. Полученные навыки могут быть полезными для решения разнообразных задач, связанных с анализом данных и машинным обучением.