

وارد می‌کند. سپس مجموعه داده `scikit-learn` و توابع مربوطه را از `NumPy`، `Pandas` این کد کتابخانه‌های لازم از جمله را بارگیری و از قبل پردازش می‌کند و یک طبقه بندی کننده نرم افزار مکس مکرر را با استفاده از صعود گرادیان دسته `MNIST` ای پیاده سازی می‌کند. طبقه بندی کننده آموزش دیده و عملکرد آن در مجموعه آزمایشی ارزیابی می‌شود. کد به خوبی ساختار یافته است و به نظر می‌رسد درست باشد.

مراحل کلیدی کد به شرح زیر است:

1. وارد شده‌اند `scikit-learn` و توابع مربوطه از `NumPy`، `Pandas`: وارد کردن کتابخانه‌های لازم.
 2. بارگیری می‌شود و به `fetch_openml` مجموعه داده با استفاده از تابع `MNIST` بارگذاری و پیش پردازش مجموعه داده. تبدیل می‌شود. مقادیر پیکسل با تقسیم بر 255 نرمال می‌شوند `NumPy` آرایه‌های
 - و `target_categories_to_numbers` تبدیل برچسب‌های هدف به اعداد و کدگذاری یک‌طرفه: توابع 3. برای تبدیل برچسب‌ها به اعداد و سپس به بردارهای کدگذاری شده یک‌هت تعریف `numbers_to_one_zero_encoding` می‌شوند.
 4. برای تقسیم مجموعه داده به 80٪ داده‌های `train_test_split` تقسیم مجموعه داده به مجموعه‌های آموزشی و آزمایشی: تابع. آموزشی و 20٪ داده‌های آزمایشی استفاده می‌شود.
 5. تعریف توابع طبقه بندی کننده سافت مکس: توابع برای پیش بینی خطی، پیش بینی سافت مکس، خروجی سافت مکس، به روز. رسانی وزن و تابع هزینه تعریف شده است.
 6. با مقادیر تصادفی مقداردهی اولیه می‌شود و طبقه بندی کننده با استفاده از `W` ماتریس وزن `softmax` آموزش طبقه بندی کننده. شیب صعودی دسته ای برای تعداد مشخصی از دوره‌ها و اندازه دسته ای آموزش داده می‌شود.
 7. برای تست مدل آموزش دیده بر روی مجموعه تست و محاسبه دقت آن تعریف شده `test_model` تست مدل آموزش دیده: تابع. است.
 8. چاپ دقت: در نهایت دقت مدل آموزش دیده روی مجموعه تست چاپ می‌شود.
- لازم به ذکر است که عملکرد طبقه بندی کننده و فرآیند تمرین ممکن است به دلیل تصادفی بودن مقدار اولیه وزن و دانه‌های تصادفی متفاوت متفاوت باشد.