روش KNN:

در این روش برای تشخیص کلاس یک داده‌ی ورودی (تست) ابتدا فاصله‌ی آن را با تمام داده‌های قبلی (آموزشی) محاسبه میکنیم و از بین این مقایسه‌ها K تای نزدیک‌ترین را انتخاب می‌کنیم. پس از آن کلاس داده را برابر با کلاسی میگیریم که در بین این K داده بیشترین تعداد را دارد. برای مثال اگر برای یک داده‌ی ورودی ۳ کلاس زیر به عنوان نزدیک ترین‌ها بدست آمده باشد (K را در اینجا ۳ فرض میکنیم)، این الگوریتم کلاس داده‌ی ورودی را TRUE بر میگرداند.

TRUE,
FALSE,
TRUE

نحوه‌ی خواندن فایل ورودی:

در ماژول dataset در ابتدا یک تابع جهت خواندن فایل قرار دارد که خط به خط فایل را خوانده و پس از آن روی کاما split میکند. در نهایت داده‌ها را به صورت لیستی از tupleها بر می‌گرداند.

نحوه‌ی ساخت داده‌های تست و آموزش:

تابع دیگر این کلاس تابع ساخت داده‌های تست و آموزش است که در آن پس از دریافت دیتاست ورودی (کل داده‌ها)، به تعداد ۲۰ درصد ورودی عدد رندم بین ۰ تا تعداد داده‌های ورودی تولید میکند، و آن سطر را در لیست داده‌های تست قرار می‌دهد. تمام سطرهای باقی‌مانده در داده‌های آموزش قرار میگیرند و در نهایت با استفاده از تابعی جدا میتوان آن‌ها را در یک فایل ذخیره کرد.

نحوه‌ی کار الگوریتم KNN:

در ماژول KNN تابع classify قرار دارد که داده‌ی ورودی را با تک تک داده‌های training dataset فاصله‌ی اقلیدسی را حساب میکند و به صورت نزولی در یک آرایه ذخیره می‌کند. پس از آن در بین k داده‌ی اول کلاسی که بیشترین تعداد را دارد را به عنوان خروجی چاپ می‌کند.

نحوه‌ی انتخاب بهترین K:

این کار را با تست کردن تمام داده‌های تست انجام میدهیم. به صورتی که تک تک آن‌ها را به تابع classify میدهیم و خروجی را بررسی میکنیم. در نهایت برای هر K درصد خطا از فرمول زیر بدست می‌آید:

Error rate = Correct Prediction / Total Test Data

با مینیمام گرفتن از این داده‌ها میتوان بهترین K را بدست آورد.