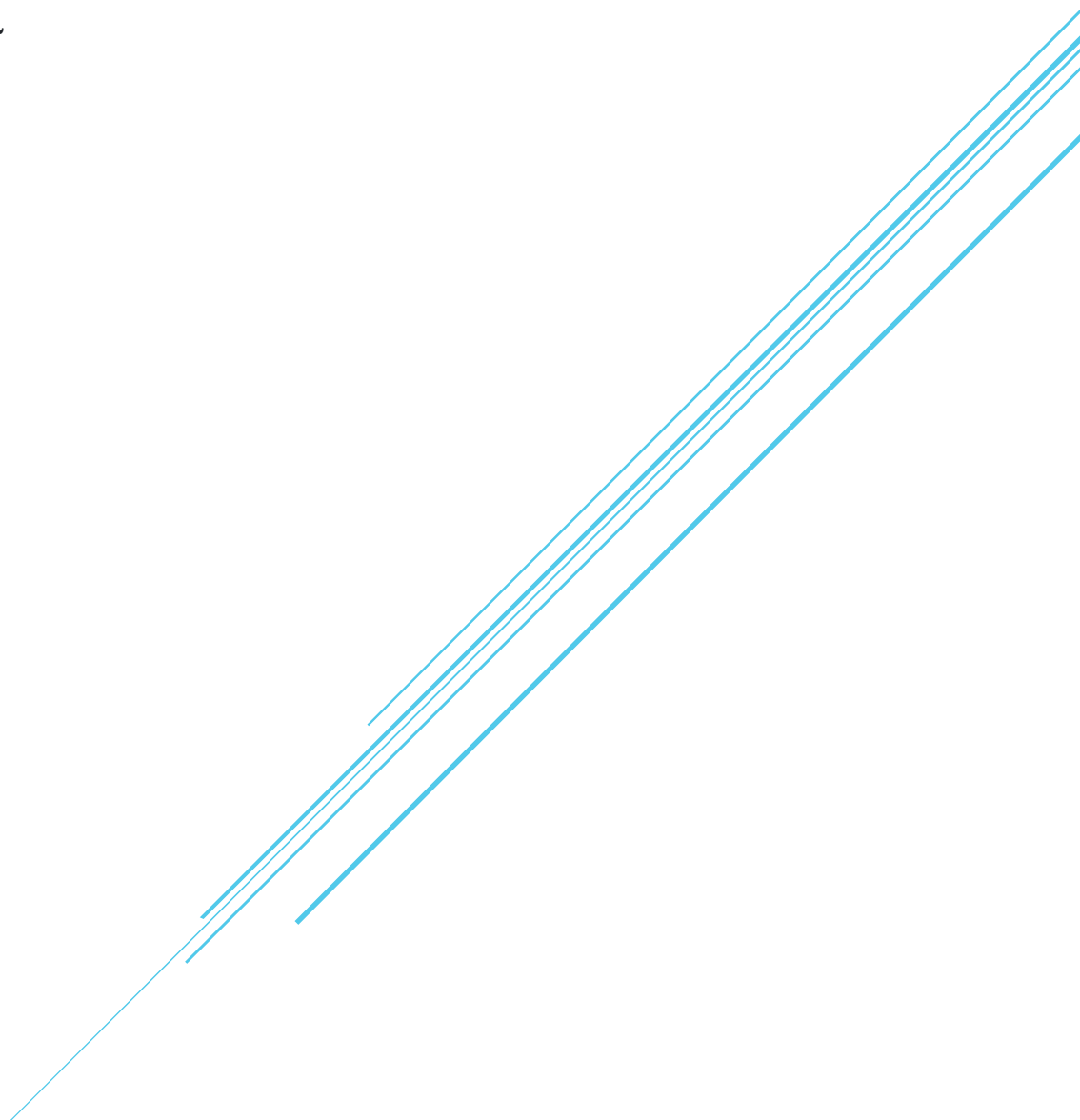


گزارش

اسم درس

۱۴۰۱/۱۰/۲۳



- نام و نام خانوادگی :
- نام استاد :
- گروه :

گزارش کار ۱

توضیح اجرای برنامه:

برای اجرای برنامه میتوان از دو تست کیس های تعبیه شده استفاده کرد. سعی شد از مثال های متنوع استفاده شود.

۱. اجرای `run_tests_blobs`

۲. اجرای `run_tests_circle`

کتابخانه های مورد نیاز:

۱. `numpy` : توابع ریاضی مورد نیاز

۲. `seaborn` : برای تجسم داده ها

۳. `Scikit-learn` : برای مدل ها و پیش بینی ها

۴. `matplotlib` : ترسیم توابع به صورت گرافیکی

• تمامی کتابخانه از طریق `pip` قابل نصب هستند.

متغیر های مهم:

۱. `NUMBER_OF_PARTITIONS` :

تعداد گروه های مورد استفاده

۲. `FIG_SIZE` :

طول و عرض سگمنت های ساخته شده در هر نمودار

تست کیس های blobs:

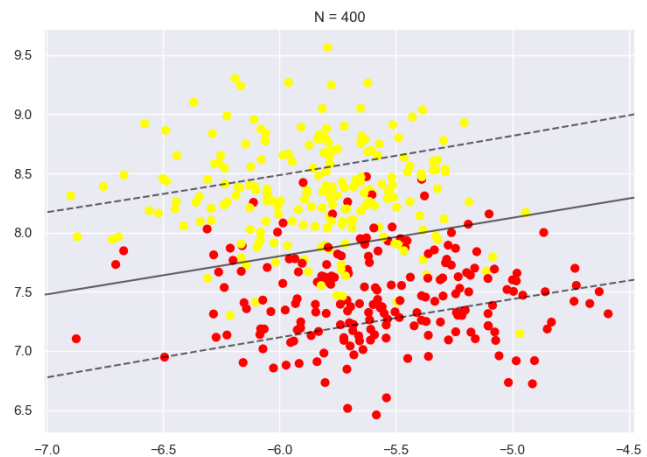
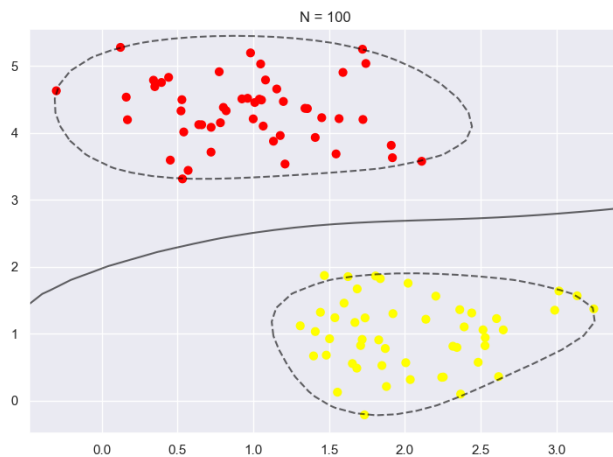
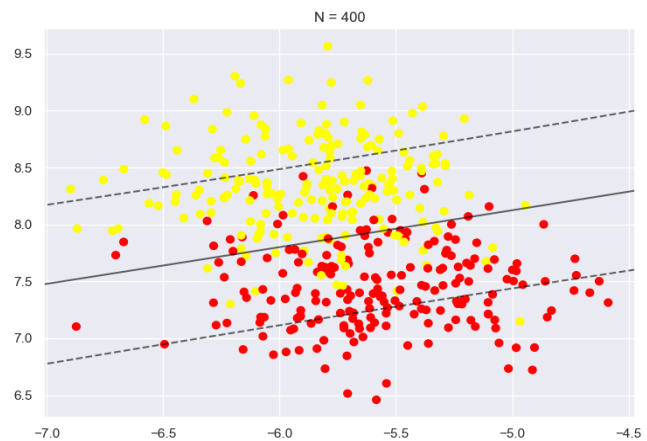
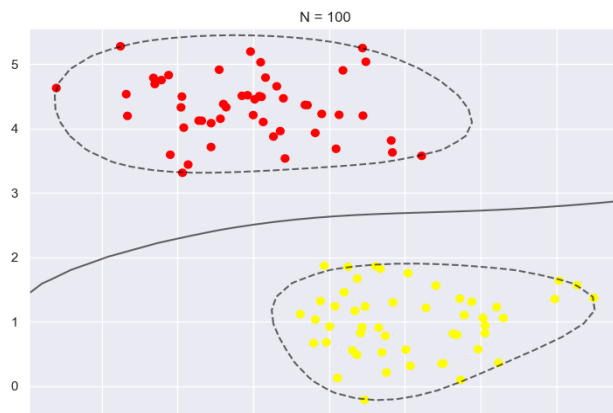
برنامه به حالتی نوشته شده که پارامترهای توابع به صورت دیکشنری هایی در یک لیست هستند. در صورتی که kernel به صورت پیش فرض باشد، میتوان از gamma استفاده کرد. در صورتی که از kernel – poly استفاده شود میتوان از degree نیز استفاده کرد.

تعداد sample ها ۱۰۰ یا ۴۰۰ است.

در هر تست هر چه میزان تست پیچیده تر باشد، نتایج ممکن است متفاوت باشند.

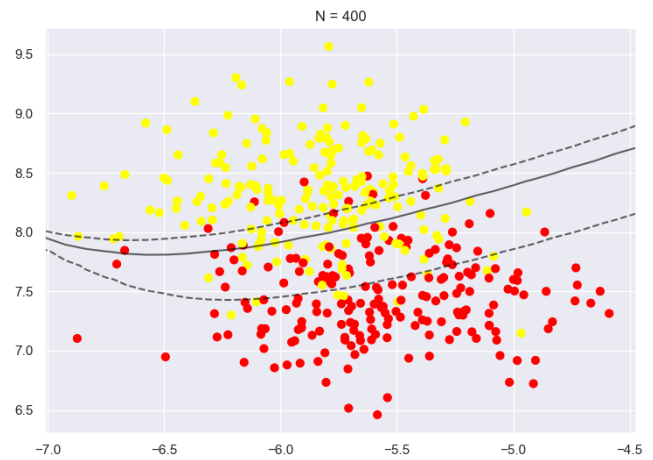
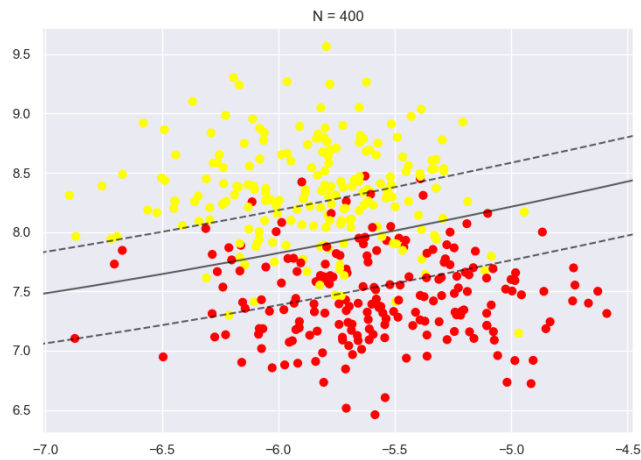
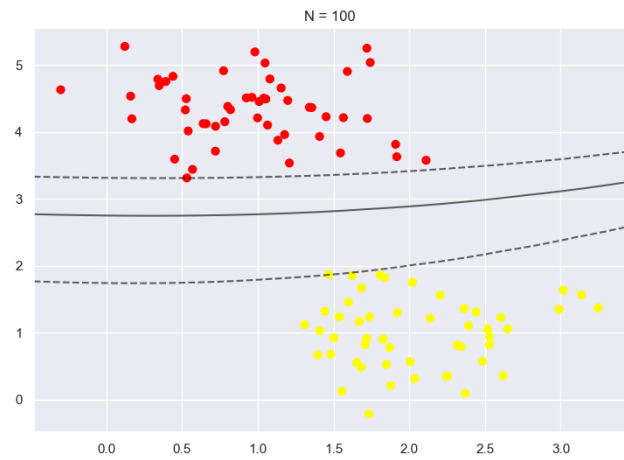
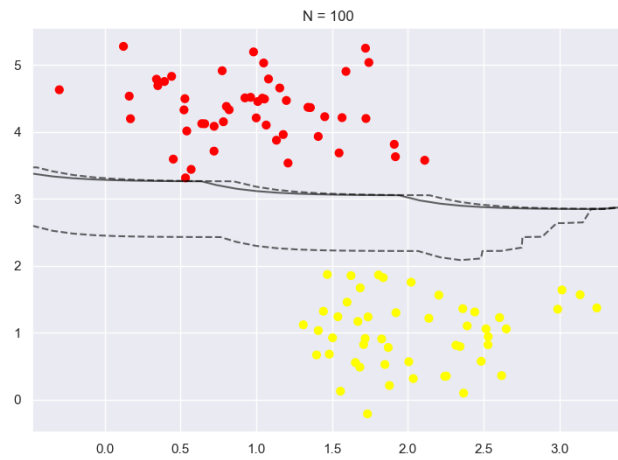
کرل rbf:

- اگر گاما auto باشد، دور هر کدام margin میکشد.
- در صورتی که اگر گاما scale باشد، شبیه کرل linear میشود.



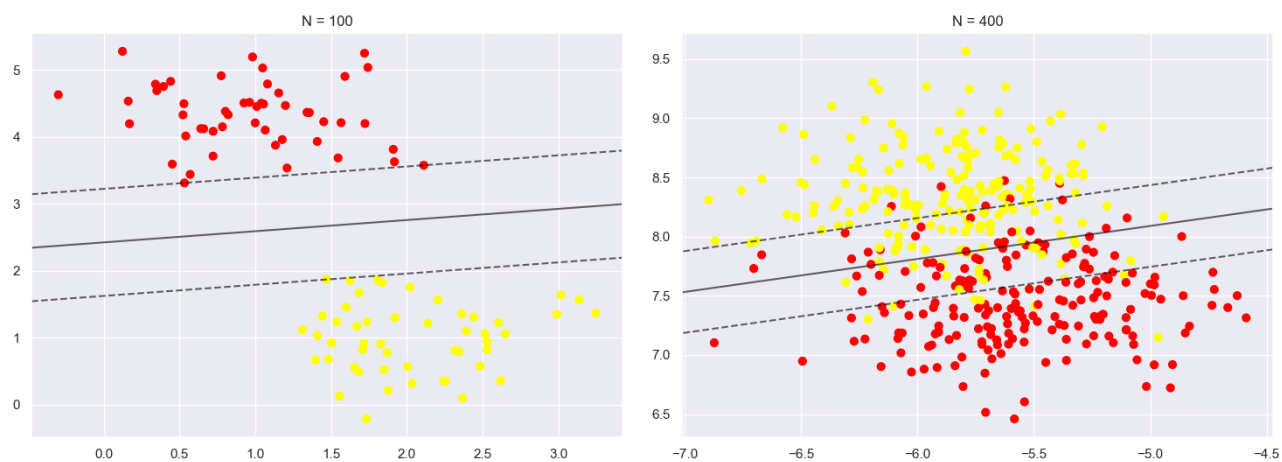
کرنل poly:

- میزان degree در میزان دقت آن دخالت دارد ولی دلیل نمیشود هر چه degree بیشتر باشد ، دقت آن نیز بهتر باشد.



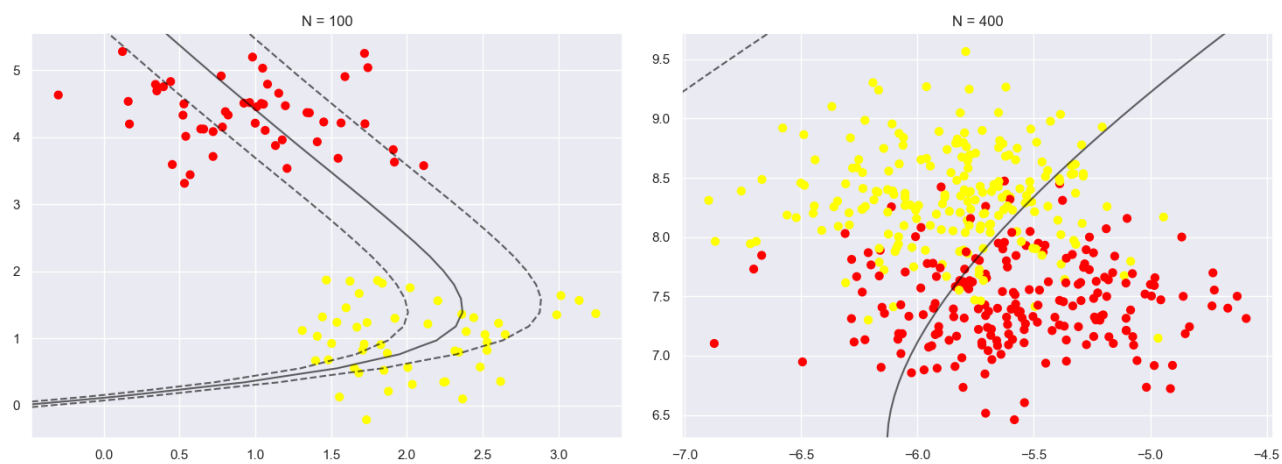
کرنل linear:

- هر چه میزان random_state بیشتر باشد ، داده ها بهم نزدیک تر میشوند.



کرنل sigmoid:

- که نتیجه ی جالبی با تست های مکرر نداد و خارج ترین حالت ممکن بود.



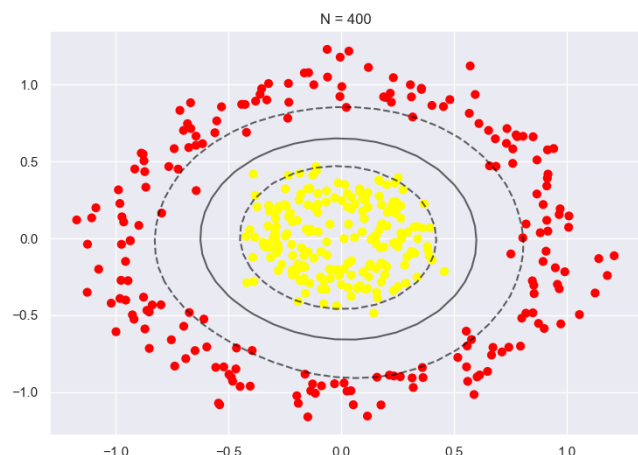
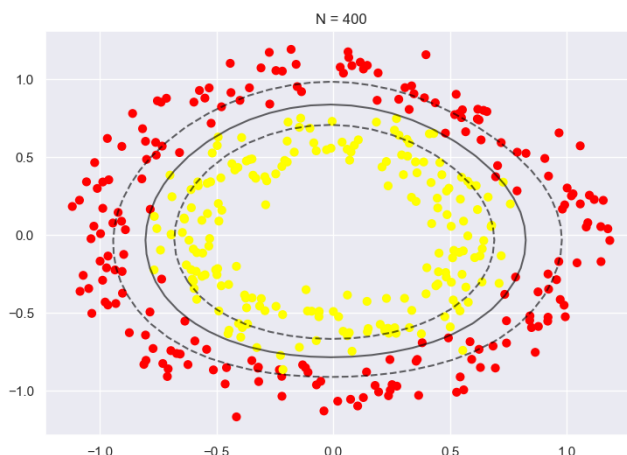
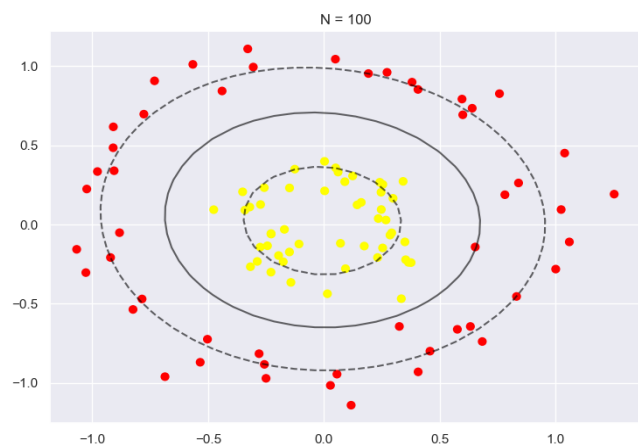
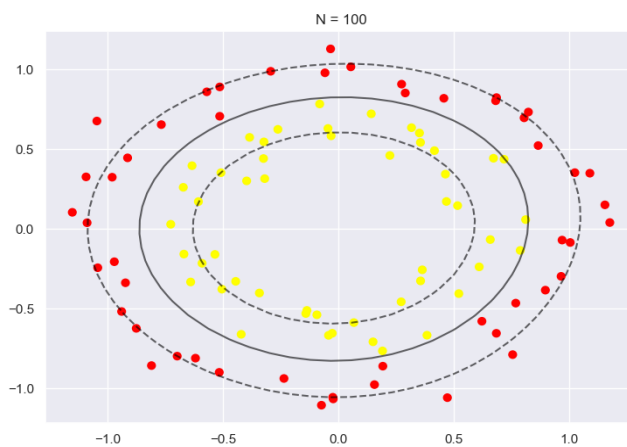
تست کیس های circle:

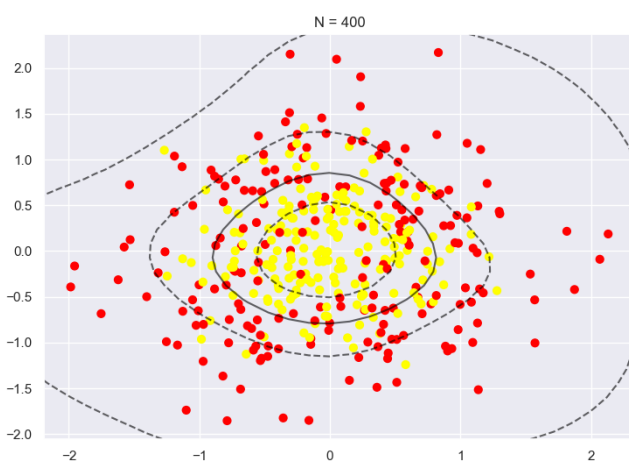
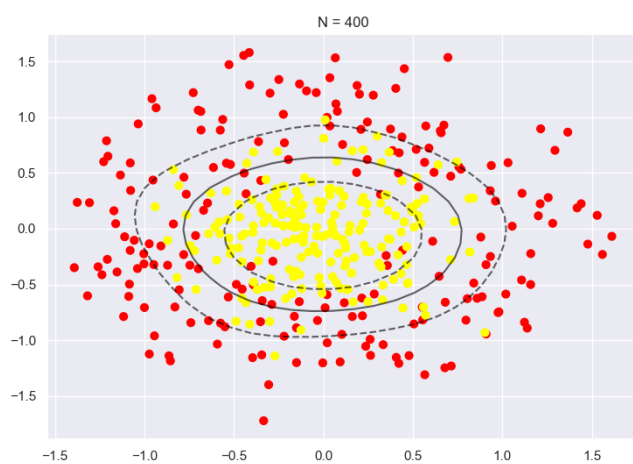
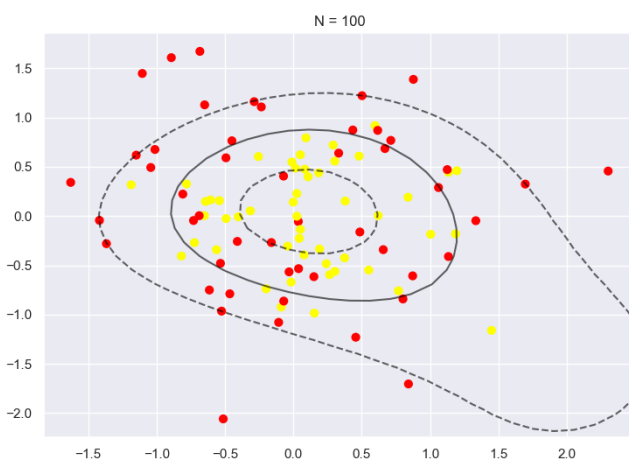
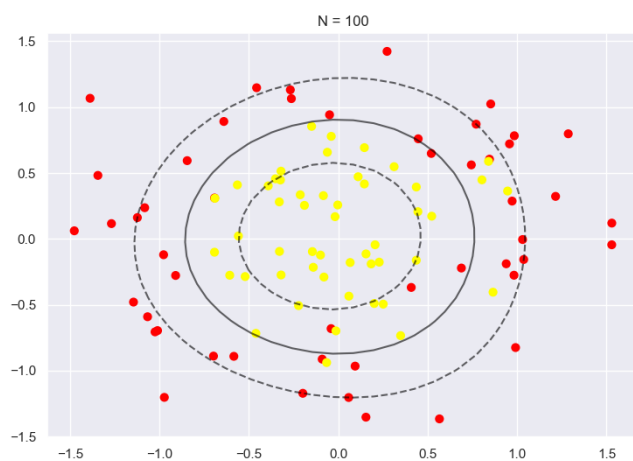
برنامه به حالتی نوشته شده که پارامترهای توابع به صورت دیکشنری هایی در یک لیست هستند. در صورتی که kernel به صورت پیش فرض باشد، میتوان از gamma استفاده کرد. در صورتی که از kernel – poly استفاده شود میتوان از degree نیز استفاده کرد. بعضی پارامترها shuffle هم دارند که در نتایج آنچنان تاثیری نداشت و فقط برای تست هستند. تعداد sample ها ۱۰۰ یا ۴۰۰ است.

در هر تست هر چه میزان تست پیچیده تر باشد، نتایج ممکن است متفاوت باشند.

کرل rbf:

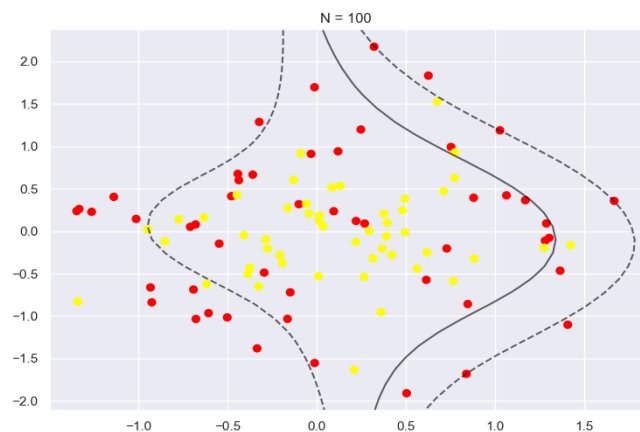
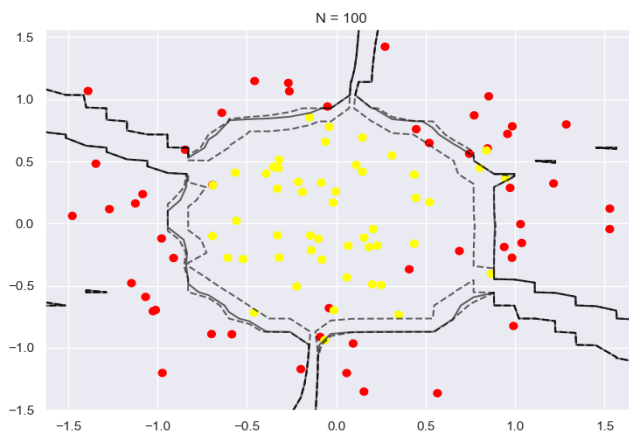
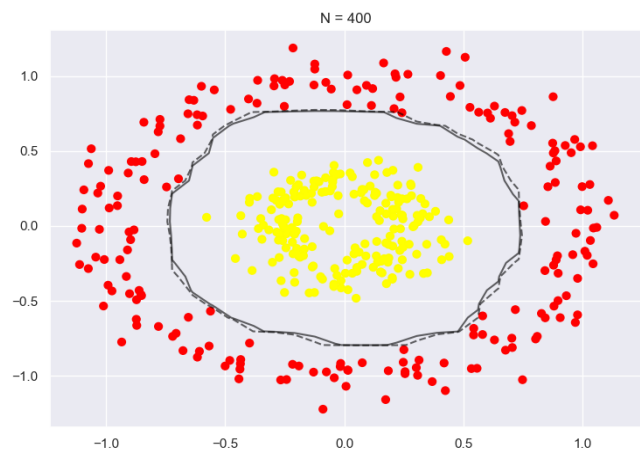
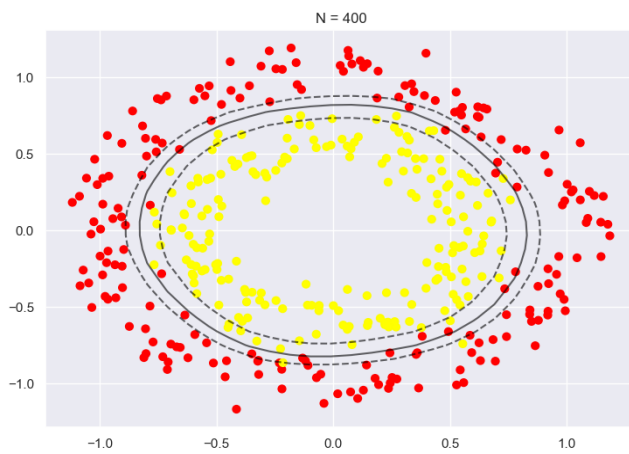
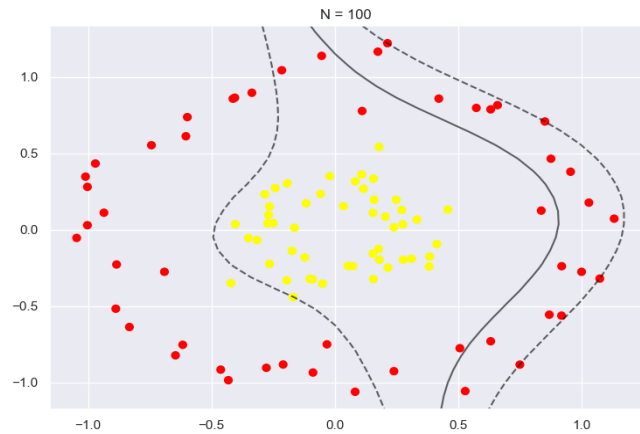
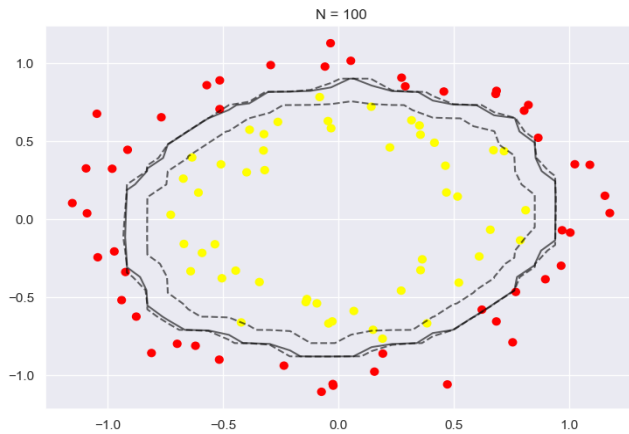
- هر چه random_state بیشتر باشد، داده ها بیشتر در هم آمیخته هستند.
- هر چه noise بیشتر باشد، نتایج به شدت افت پیدا میکنند و پیش بینی کم کم از بین میرود.





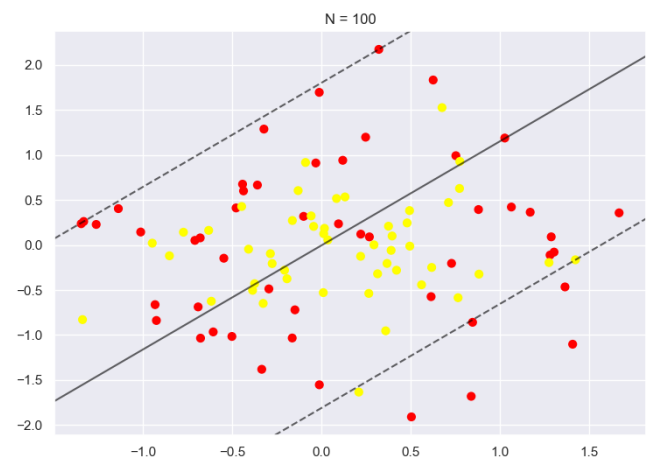
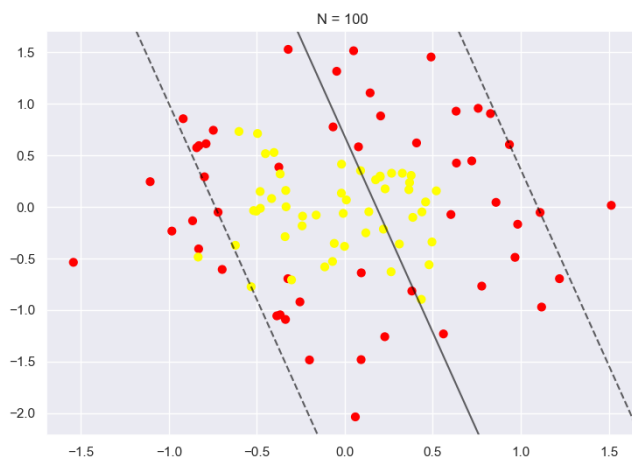
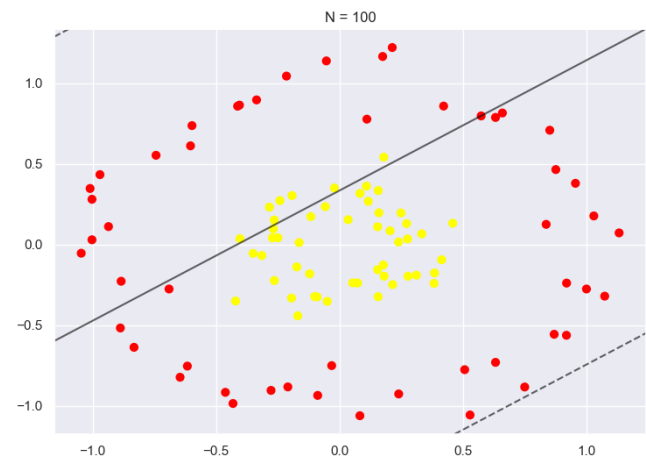
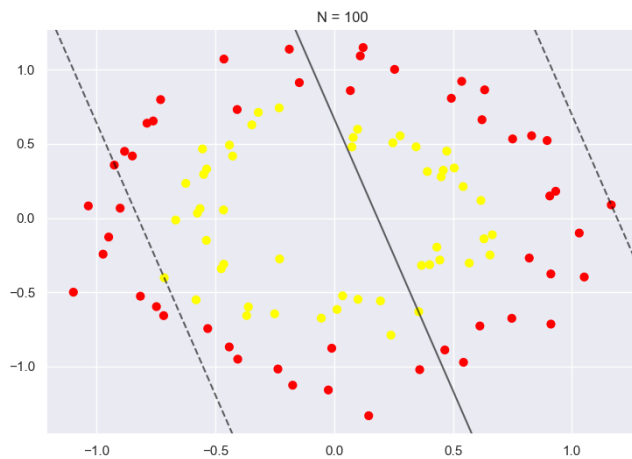
کرل poly:

- هر چه میزان factor و degree بیشتر میشود ، نتایج بهتری میگیرد.
- هر چه noise بیشتر باشد ، نتایج به شدت افت پیدا میکنند و پیش بینی کم کم از بین میرود.



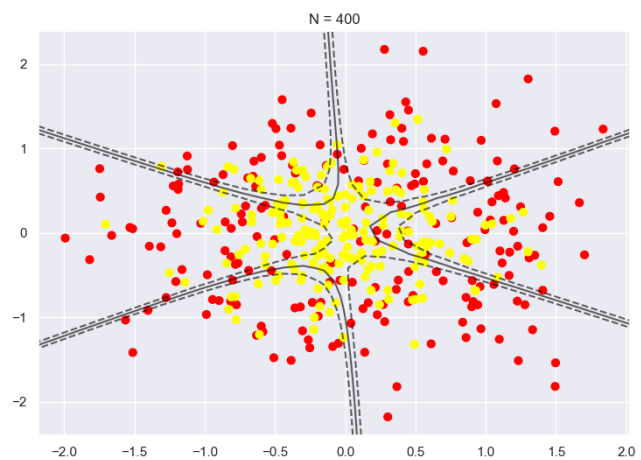
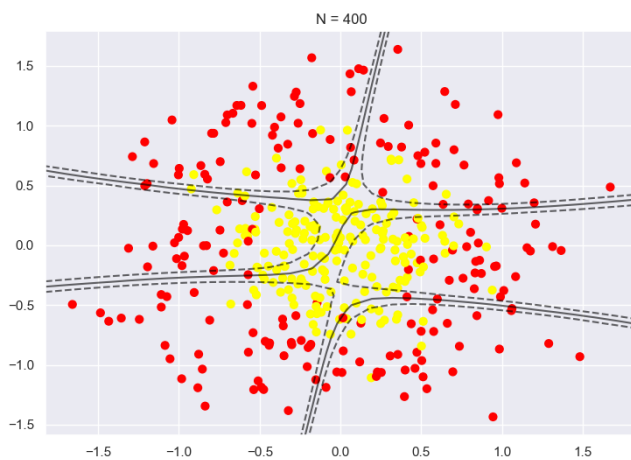
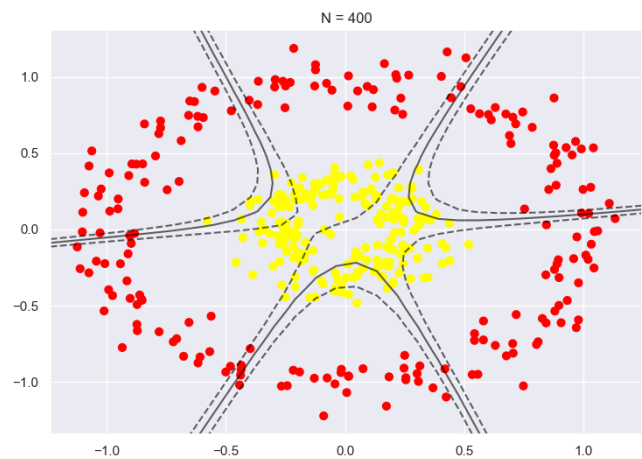
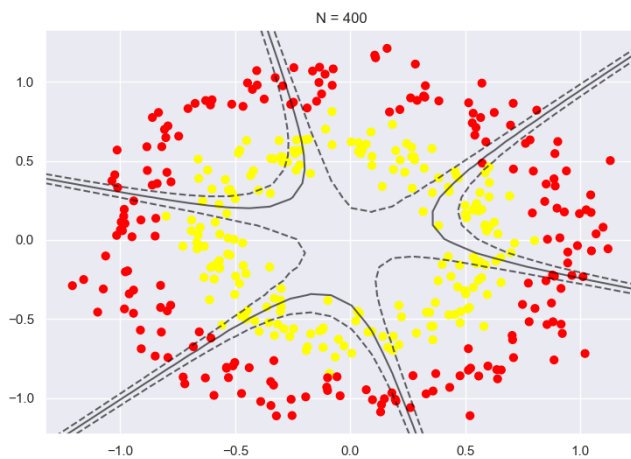
کرنل linear:

- در این نوع مثال ها خوب جواب نمیدهد.
- هر چه مقادیر تغییر کنند ، وضع بهتری را ارائه نمیدهند.



کرنل sigmoid:

- در این نوع مثال ها خوب جواب نمیدهد.
- هر چه مقادیر تغییر کنند ، وضع بهتری را ارائه نمیدهند.
- بعضی شکل ها بسیار عجیب هستند.



گزارش کار ۲

توضیح اجرای برنامه:

برای اجرای برنامه باید تصاویر در پوشه ی USPS_images/images/ به دو حالت

۱. Train

۲. Test

کتابخانه های مورد نیاز:

- ۱. numpy : توابع ریاضی مورد نیاز
 - ۲. pandas : برای آنالیز دیتا ها
 - ۳. Scikit-learn : برای مدل ها و پیش بینی ها
 - ۴. scikit-image : برای پراسس کردن عکس ها
- تمامی کتابخانه از طریق pip قابل نصب هستند.

متغیر های مهم:

۱. CATEGORIES :

عدد هایی که باید پردازش شوند.

۲. TRAIN_DIR :

مسیر پوشه ای که فایل های train در آن قرار دارند.

۳. TEST_DIR :

مسیر پوشه ای که فایل های test در آن قرار دارند.

نتایج برنامه:

- برنامه به حالتی نوشته شده که پارامترهای توابع به صورت دیکشنری هایی در یک لیست هستند.
- تعداد داده های train شده هر چه بیشتر باشد ، نتایج بهتر است اما زمان train شدن به شدت بالا میرود.
- در هر تست هر چه میزان تست پیچیده تر باشد ، نتایج ممکن است متفاوت باشند.
- به صورت کلی فایل به دست آمده از کرنل rbf حجم بیشتری دارد.
- به صورت کلی زمان اجرای کرنل rbf از همه بیشتر است.
- به صورت کلی نتایج به دست آمده از کرنل linear از دیگر کرنل ها بهتر است.
- در صورتی که تعداد اعضای لیست gamma بیشتر شود ، نتایج به شدت کند میشوند.
- در صورتی که تعداد اعضای لیست C (دقت) بیشتر شود ، نتایج به شدت کند میشوند.
- میتوان همزمان از دو کرنل استفاده کرد ولی دلیلی بر نتایج بهتر نمیشود.
- میزان train کردن به صورت پیش فرض ۱۰ در نظر گرفته شده است. در این صورت برای هر تست حدود ۲ الی ۴ دقیقه زمان میبرد. میتوان عدد را بیشتر کرد ولی زمان برنامه به صورت نمایی بالا میرود.

برخی نتایج در فایل **result.txt** قابل مشاهده هستند.