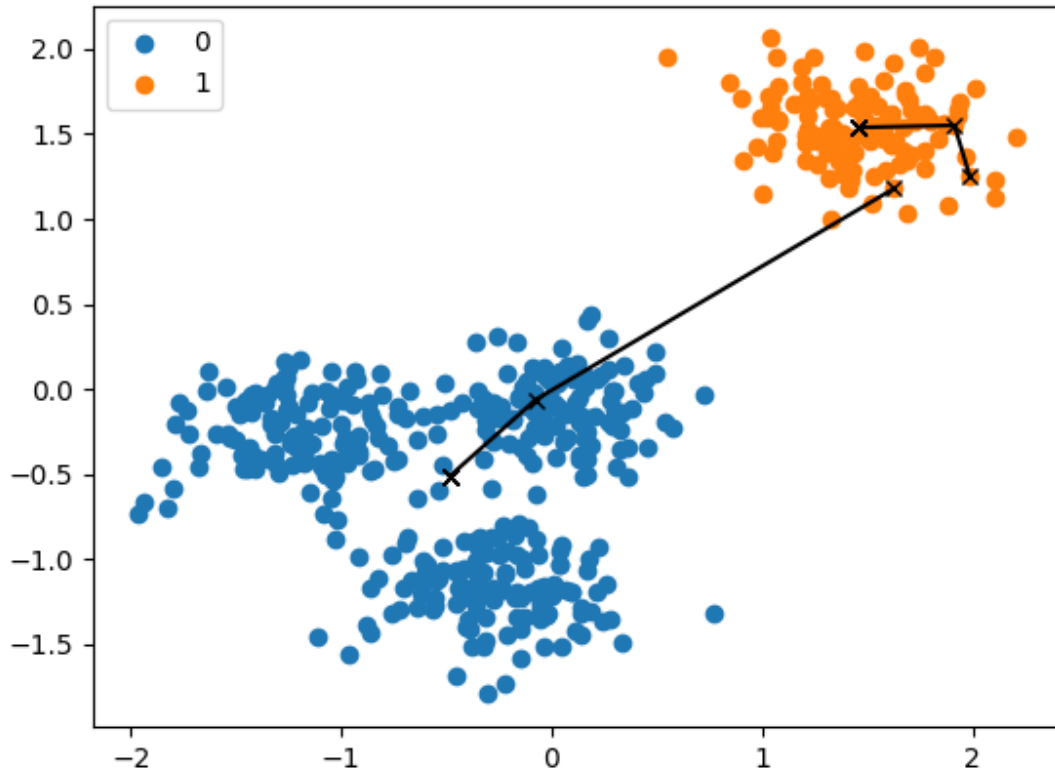


## سوال ۱

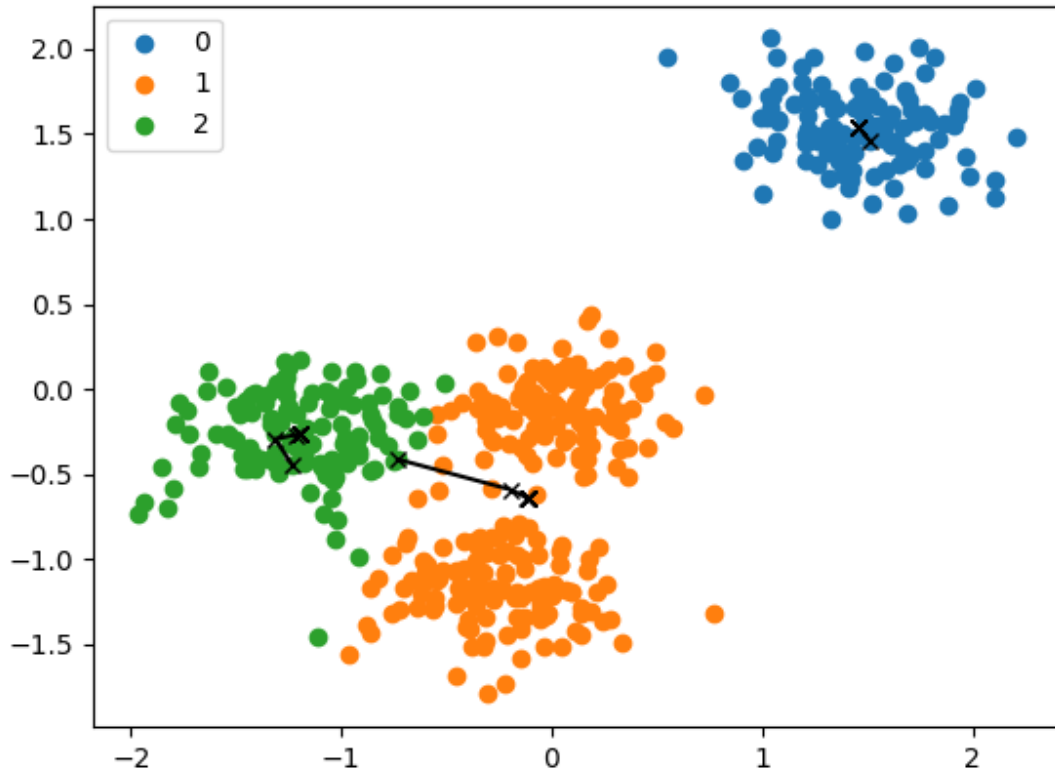
برای این بخش توابع `initialize_centers` و `find_closest_centers` و `compute_means` پیاده سازی شده‌اند و همچنین تعداد پیمایش‌ها ۱۵ قرار داده شده است. تابع `initialize_centers` به تعداد ورودی داده شده مرکز از داده‌ها به صورت اتفاقی و با شانس برابر انتخاب می‌کند. تابع `find_closest_centers` برای هر داده نزدیک‌ترین مرکز را بر حسب فاصله‌ی اقلیدسی پیدا می‌کند. تابع `compute_means` مرکز هر خوشه را به صورت میانگین نقاط موجود در آن خوشه محاسبه می‌کند. خروجی برای هر `k` به شرح زیر است.

خروجی برای  $k = 2$ :



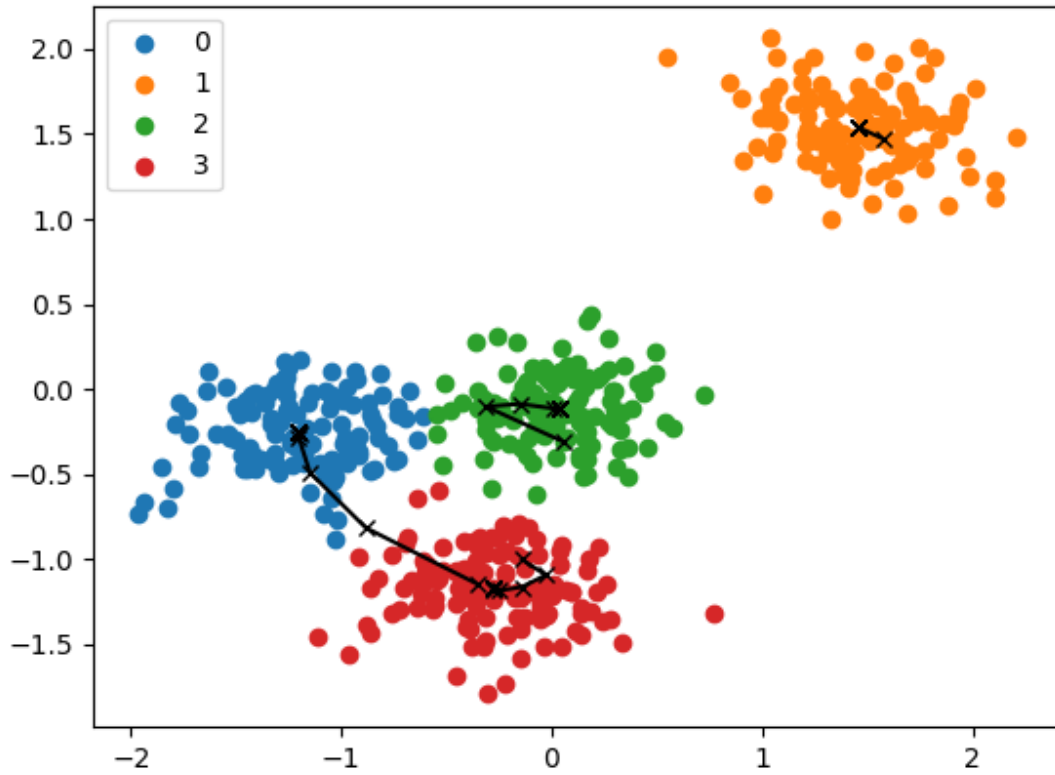
امیرحسین پاشایی هیر ۹۷۳۱۰۱۳

خروجی برای  $k = ۳$ :



امیرحسین پاشایی هیر ۹۷۳۱۰۱۳

خروجی برای  $k = 4$ :



## سوال ۲

برای این قسمت دو تابع پیاده سازی شده است که به شرح زیر است.

تابع اول، تابع `pca` است، در این تابع ابتدا ماتریس داده‌ها ( $D$ ) را می‌سازیم و سپس با کم کردن میانگین کل داده‌ها از هر داده ماتریس  $D'$  را می‌سازیم. سپس با میانگین گرفتن ماتریس حاصل از ضرب هر داده در ترانهاده شده‌ی آن ماتریس  $\Sigma$  را می‌سازیم. سپس با استفاده از کتابخانه‌ی `numpy` و تابع `numpy.linalg.eig` بردارها و مقادیر ویژه ماتریس  $\Sigma$  را بدست آورده و بردارهای ویژه ماتریس را بر اساس بزرگی مقدار آن‌ها مرتب می‌کنیم. در نهایت هر داده‌ی ماتریس  $D$  را بر تعداد بعد ( $r=2$ ) بردارهای ویژه بدست آمده، تصویر کرده و داده‌های جدید را بدست می‌آوریم.

تابع دوم، تابع `svm` است، که در آن ابتدا با دستور `train_test_split` داده‌ها را به دو بخش آموزشی و تستی تقسیم کرده و با استفاده از `SVC` در کتابخانه‌ی `sklearn` بر روی داده‌های آموزشی یک مدل `SVM` را `fit` می‌کنیم، سپس داده‌های تستی را با مدل `SVM` ساخته شده `predict` می‌کنیم و دقت را بر اساس تعداد درست حدس‌های زده شده توسط مدل اندازه گیری می‌کنیم. نتیجه‌ی حاصل به این صورت است که دقت مدل بر روی داده‌های اصلی حدود ۹۶ درصد می‌باشد و دقت مدل بر روی داده‌های دو بعدی بدست آمده با `PCA` حدود ۸۶ می‌باشد که با توجه به اینکه تعداد ابعاد داده‌ها نصف شده است بسیار مناسب است.

همچنین شکل داده‌های دو بعدی به دست آمده توسط `PCA` به شکل زیر است:

امیر حسین پاشایی هیر ۹۷۳۱۰۱۳

