باسمه تعالی

دانشگاه صنعتی شریف

دانشکده مهندسی برق

مقدمه‌ای بر یادگیری ماشین ــ دکتر سیّد جمال‌الدین گلستانی

بهراد منیری ــ ۹۵۱۰۹۵۶۴

گزارش تمرین سری هشتم

بخش اول

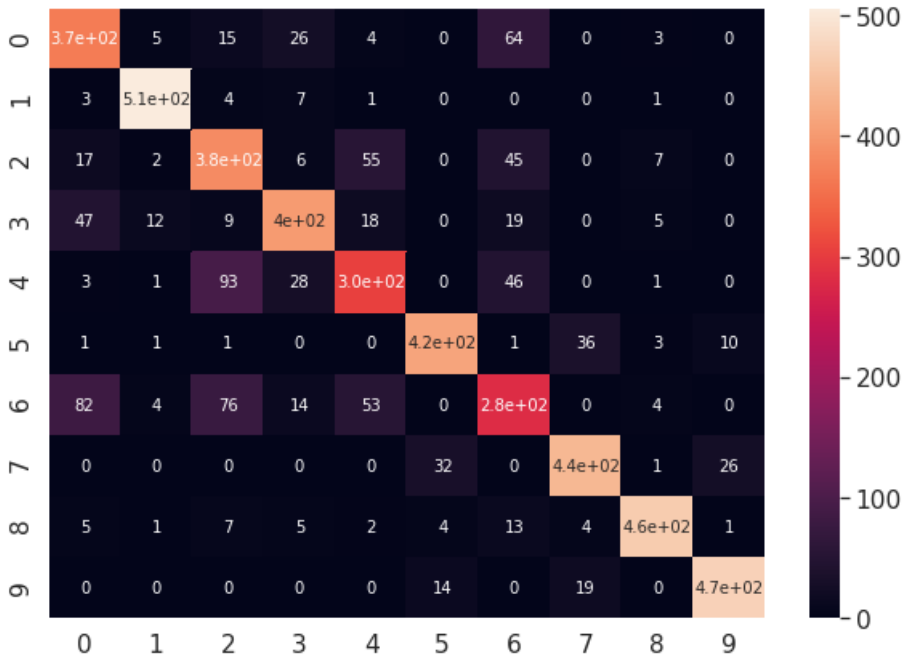
طبقه‌بندی

در این بخش به طبقه‌بندی دیتاست معروف Fashion MNIST با استفاده از چندین روش مختلف کرده و برای هر روش، پارامترهای مربوطه ره به نحوی تنظیم می‌کنیم که بهترین عملکرد ممکن را بر روی داده‌های تست داشته باشیم. نصف داده‌ها به عنوان داده‌ی آموزش و نصف دیگر به عنوان داده‌ی تست استفاده شده است. در کل ۱۰۰۰۰ تصویر در ۱۰ کلاس در اختیار داریم. در انتها نیز روش‌های استفاده‌شده را به یکدیگر مقایسه می‌کنیم. برای هر روش ماتریس کانفیوژن و همچنین درصد کلی عملکرد، آورده شده است.

۱ Linear SVM

پیاده‌سازی SVM خطی. در این بخش از تابع sklearn.SVC استفاده کرده و کرنل را خطی قرار دادیم.

درصد عملکرد کلی این روش 80.66% است.



شکل ۱: ماتریس کانفیوژن Linear SVM

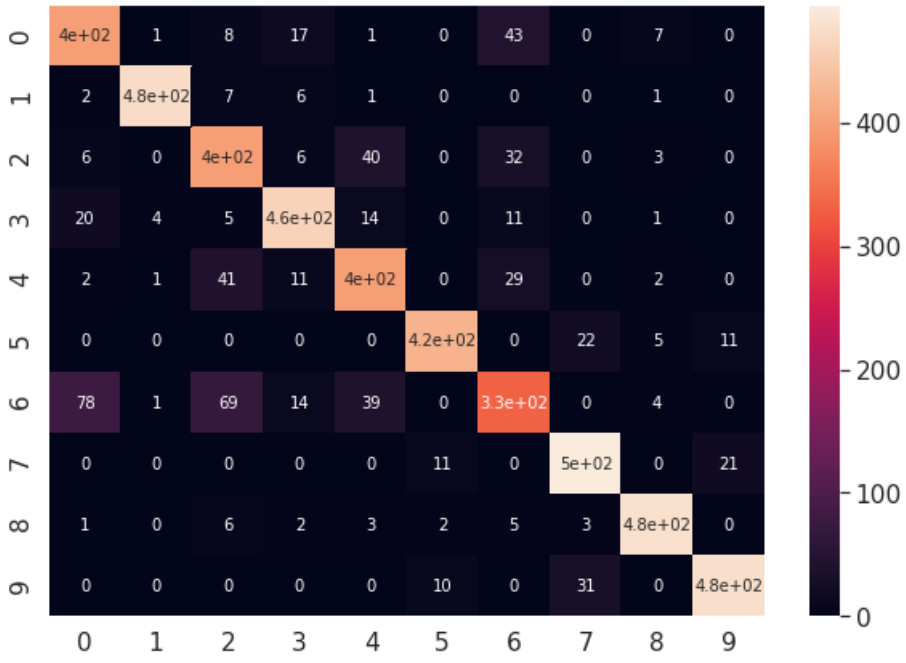
۲ Kernel SVM

پیاده‌سازی SVM با کرنل گاوسی. در این بخش از تابع sklearn.SVC استفاده کرده و کرنل را RBF قرار دادیم. همچنین

$$C = 10^6, \quad \gamma = 2 \times 10^{-7}$$

با آزمون و خطا به عنوان پارامترهای ما انتخاب شدند.

درصد عملکرد کلی این روش 86.88% است.



شکل ۲: ماتریس کانفیوژن Kernel SVM

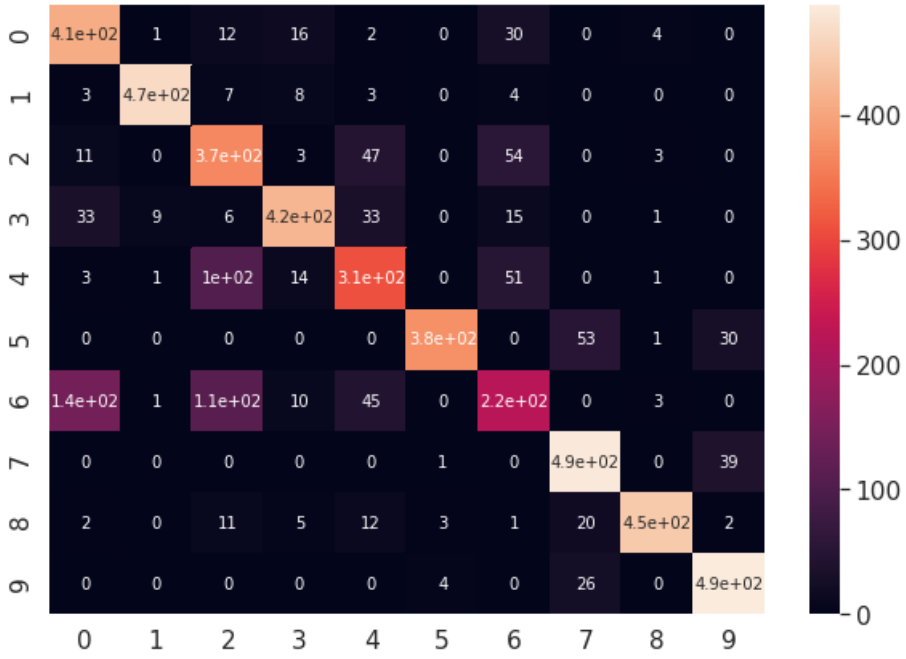
kNN ۳

در این بخش با k-Nearest Neighbor به طبقه‌بندی می‌پردازیم.

$$k = 6$$

با آزمون و خطا به عنوان پارامتر انتخاب شد.

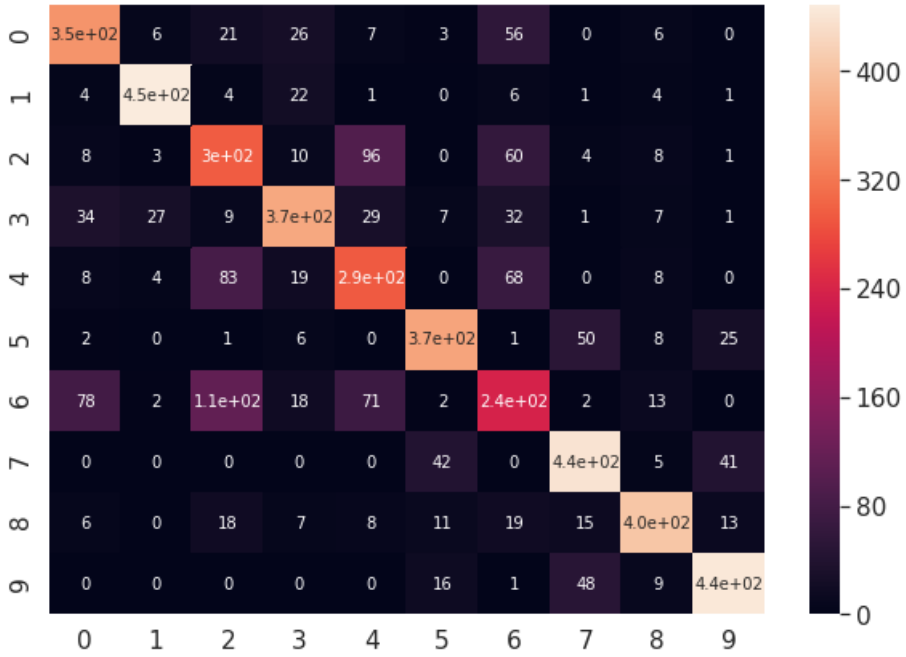
درصد عملکرد کلی این روش 79.94% است.



شکل ۳: ماتریس کانفیوژن kNN

۴ Decision Tree

در این بخش از Decision Tree برای طبقه‌بندی استفاده شده است. درصد عملکرد کلی این روش 73.06% است.

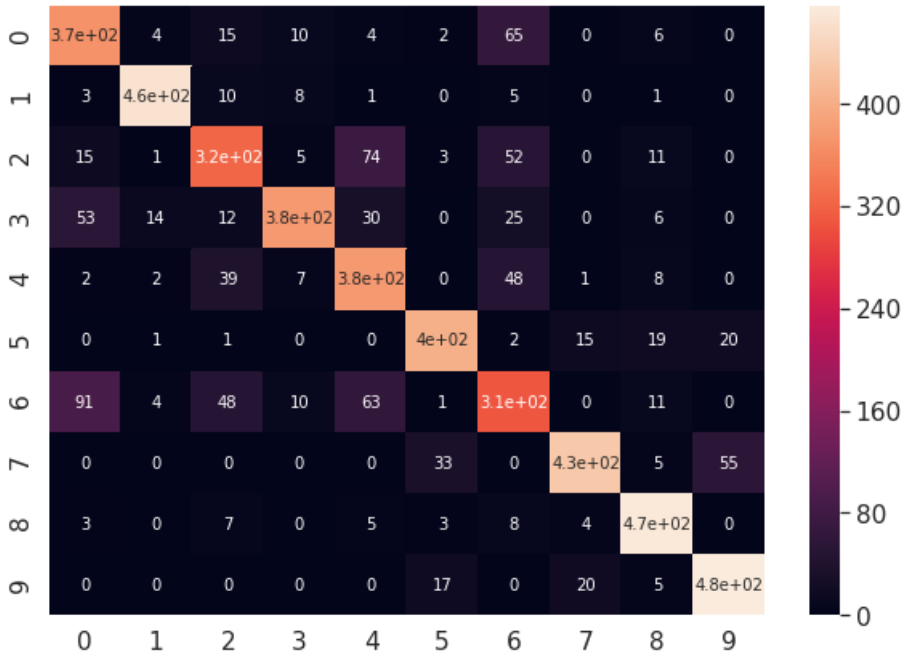


شکل ۴: ماتریس کانفیوژن Decision Tree

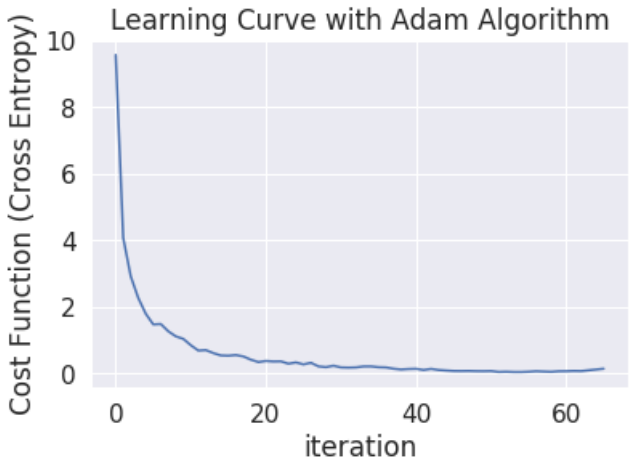
۵ MLP with Adam Optimizer

در این بخش یک شبکه‌ی Multi-Layer Preceptron با دو لایه‌ی مخفی که هر کدام ۱۰۰ نورون دارند آموزش داده می‌شود. تابع Activation این نورون‌ها ReLU انتخاب شده است ولی به‌جای SGD از الگوریتم Adam برای بهینه‌سازی استفاده شده است. تابع تلف به صورت پیش‌فرض Cross Entropy است.

درصد عملکرد این روش 80.14% است. شکل زیر، تغییرات تابع تلف را در تکرارهای الگوریتم بهینه‌سازی مشاهده می‌کنید.



شکل ۵: ماتریس کانفیوژن MLP با Adam

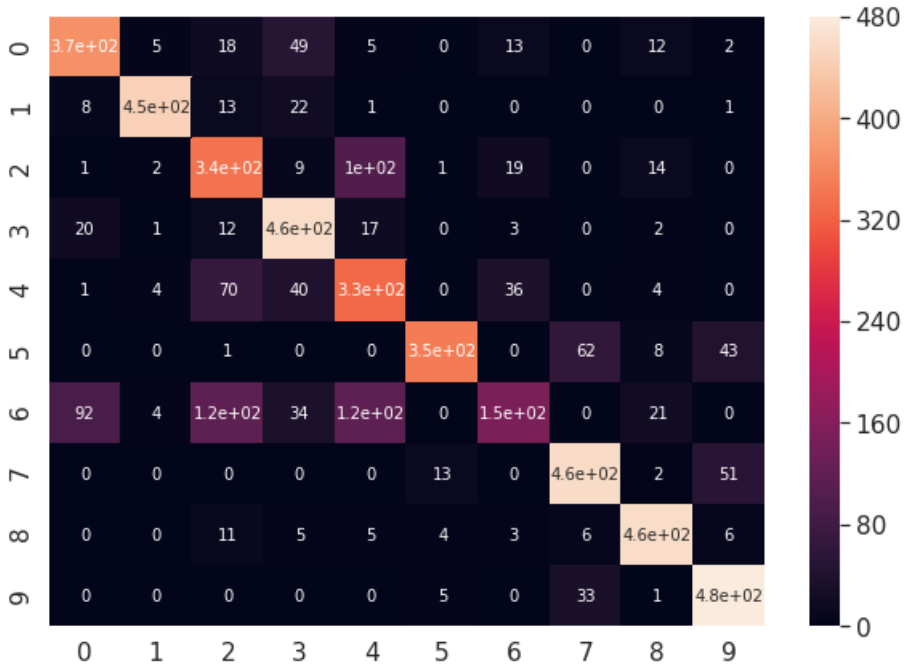


شکل ۶: Learning Curve of MLP with Adam Algorithm

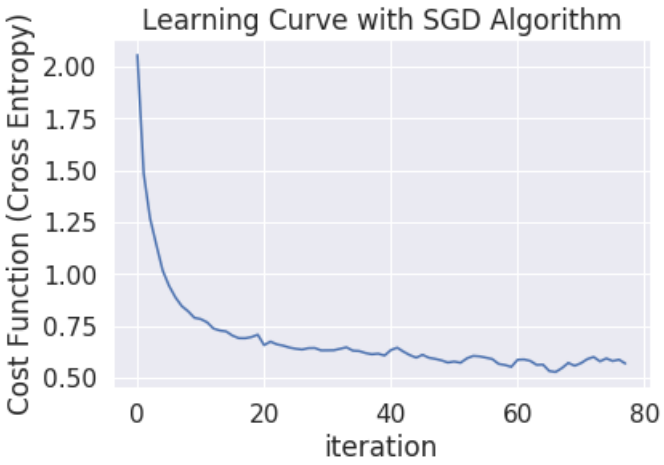
۶ MLP with SGD Optimizer

در این بخش یک شبکه‌ی Multi-Layer Preceptron با دو لایه‌ی مخفی که هر کدام ۱۰۰ نورون دارند آموزش داده می‌شود. تابع Activation این نورون‌ها tanh انتخاب شده است. برای بهینه‌سازی از الگوریتم SGD استفاده شده است. تابع تلف به صورت پیش‌فرض Cross Entropy است.

درصد عملکرد این روش 76.96% است. شکل زیر، تغییرات تابع تلف را در تکرارهای الگوریتم بهینه‌سازی مشاهده می‌کنید.



شکل ۷: ماتریس کانفیوژن MLP با SGD



شکل ۸: Learning Curve of MLP with SGD Algorithm

۷ مقایسه روش‌های طبقه‌بندی

جدول زیر نتایج طبقه‌بندی روش‌های مختلف را در کنار هم‌دیگر آورده است.

Method	Params	Accuracy
Linear SVM	-	80.66%
Kernel SVM	$C = 10 \ \ \gamma = 2 \times 10^{-7}$	86.88%
kNN	k = 6	79.94%
Decision Tree	-	73.06%
MLP (Adam)	ReLU	80.14%
MLP (SGD)	tanh	76.96%

جدول ۱: مقایسه‌ی عملکرد روش‌های طبقه‌بندی

بهترین عملکرد کلی را روش Kernel SVM داشت. این روش در مقایسه با روش های مانند kNN از سرعت یادگیری بالاتری نیز برخوردار بود. اشتباه بین لیبل صفر و شش (پیراهن و تی‌شرت) بسیار شایع بود در حالی که این اشتباه تا حد زیادی در روش Kernel SVM حل شده بود. این روش به شدت به پارامتر γ وابسته است و مقدار آن با دقت فراوانی تنظیم شده است.

در یادگیری شبکه‌عصبی، مشاهده شده که استفاده از الگوریتم Adam به جای SGD باعث می‌شود به نقطه‌ی بهینه‌تری برسم فلذا نتایج این روش نیز در گزارش آمده است.

بخش دوم

خوشه‌بندی

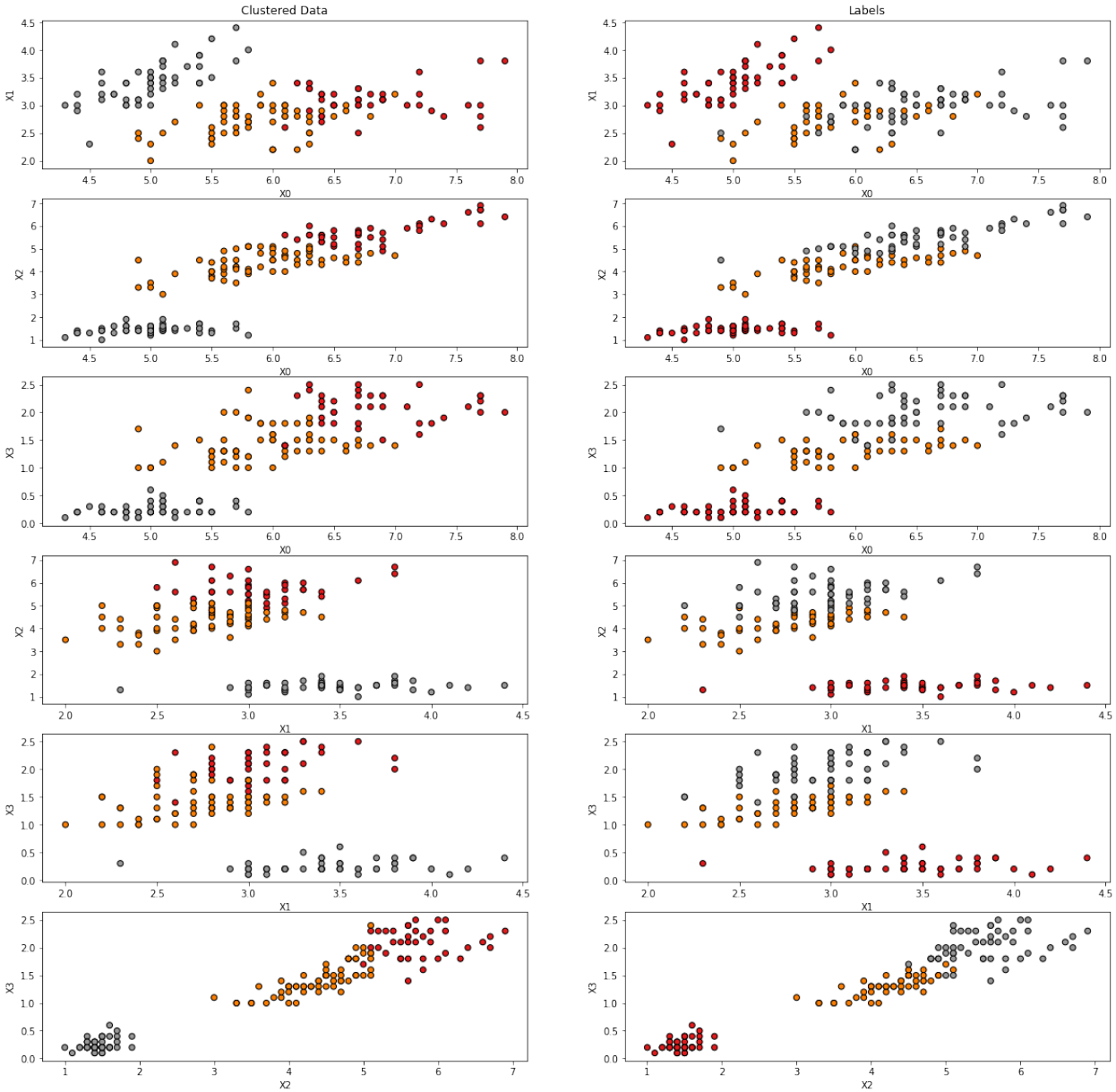
۸ پیاده‌سازی K-Means

در ابتدا به پیاده‌سازی الگوریتم K-Means می‌پردازیم. این الگوریتم به این نحو عمل می‌کند که در ابتدا نقاطی به صورت تصادفی به عنوان مرکز خوشه‌ها انتخاب می‌وند و سپس به صورت تکراری، یک نقطه تصادفی انتخاب می‌شود و فاصله‌ی آن با مراکز دسته‌ها سنجیده می‌شود و در صورتی که دسته‌ای وجود داشته باشد که مرکز آن به این نقطه از مرکز دسته‌ای که هم‌اکنون نقطه در آن قرار دارد نزدیک‌تر باشد، نقطه به دسته‌ی مذکور اضافه می‌شود و سپس مرکز دسته‌ها بروزرسانی می‌شوند. پیاده‌سازی این الگوریتم بر پایه‌ی numpy در پایتون انجام شد.

۹ خوشه‌بندی iris

در این بخش به خوشه‌بندی دیتاست معروف iris می‌پردازیم. در این مرحله، از هر چهار ویژگی برای خوشه‌بندی استفاده می‌کنیم. برای خوشه‌بندی از K-Means استفاده می‌کنیم که در آن $k = 3$ قرار دادیم.

نکته‌ی بسیار جالب این است که این خوشه‌بندی بسیار شبیه به لیبل‌های واقعی نوع گل است در حالی که این نحوه‌ی یادگیری unsupervised است. شکل (۹) خروجی این خوشه‌بندی (سمت چپ) و همچنین لیبل‌های واقعی (سمت راست) را نشان می‌دهد. تطابق زیادی بین هر دو نمودار دیده می‌شود. در حال که لزومی نداشت این اتفاق بیافتد.



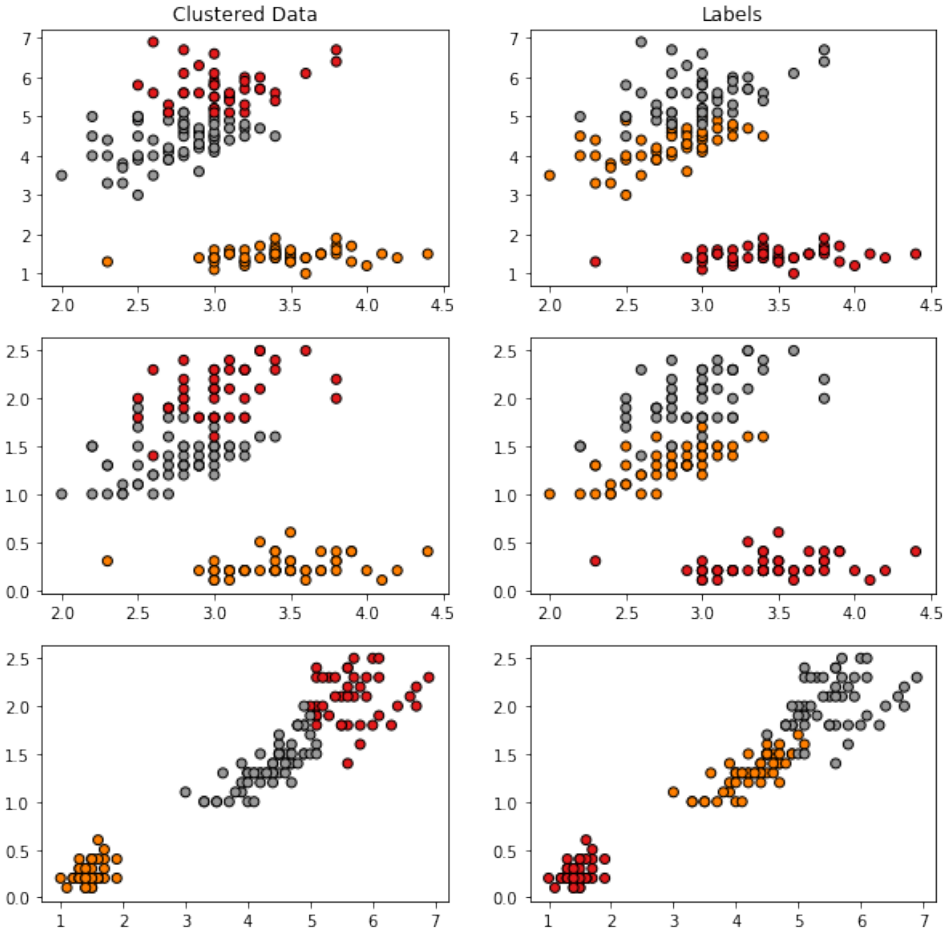
شکل ۹: برسی خوشه‌بندی ــ نمودار سمت راست بر حسب لیبل‌های واقعی و نمودار سمت چپ بر اساس خوشه‌بندی است

۱۰ حذف یک ویژگی

در این بخش قصد داریم یک ویژگی را حذف کنیم و بررسی کنیم که آیا حذف این ویژگی تاثیر معناداری بر خوشه‌بندی ما دارد یا خیر. هر بار یکی از ویژگی‌ها را حذف کرده و نتایج خوشه‌بندی را در این گزارش درج می‌کنیم.

۱.۱۰ حذف Sepal Width

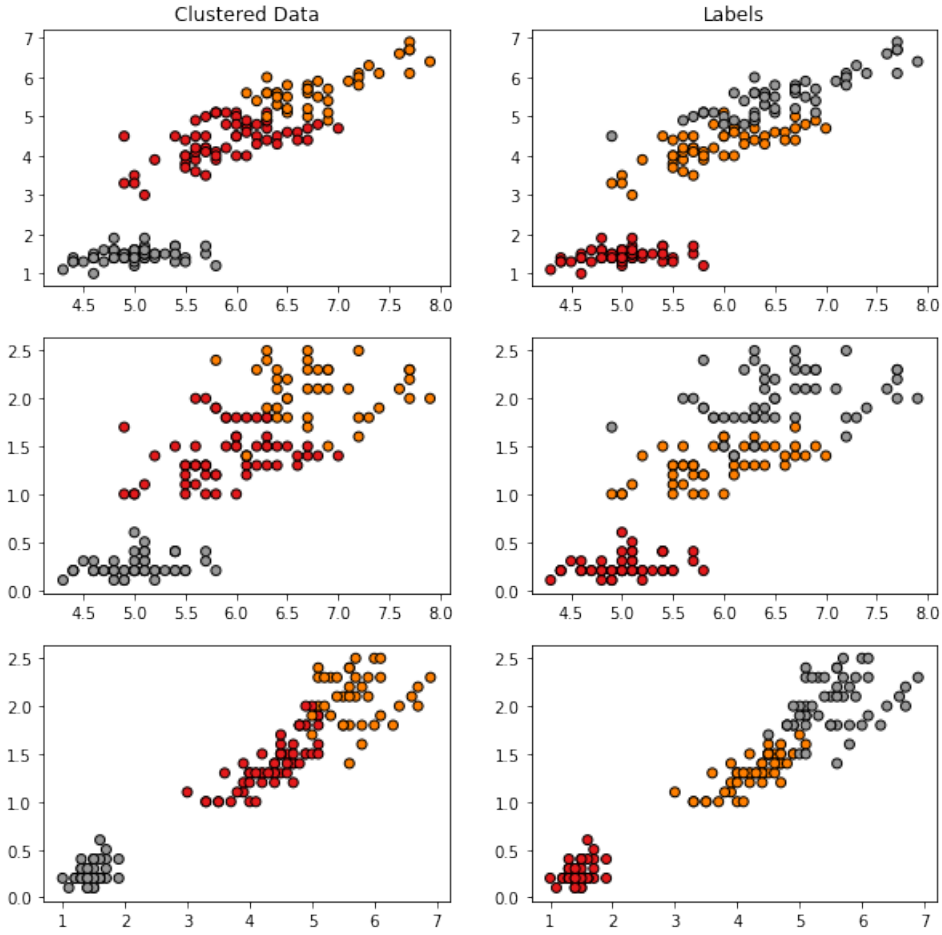
در این بخش ویژگی Sepal Width را حذف و سپس خوشه‌بندی می‌کنیم. در شکل (۱۰) نمودار های سمت چپ مربوط به نتایج خوشه‌بندی و نمودارهای سمت راست مربوط به لیبل‌های واقعی گل‌هاست. صحت خوشه‌بندی تغییر محسوسی نیافته است.



شکل ۱۰: بررسی خوشه‌بندی ـ نمودار سمت راست بر حسب لیبل‌های واقعی و نمودار سمت چپ بر اساس خوشه‌بندی است. در این‌جا ویژگی Sepal Width حذف شده است.

۲.۱۰ حذف Sepal Length

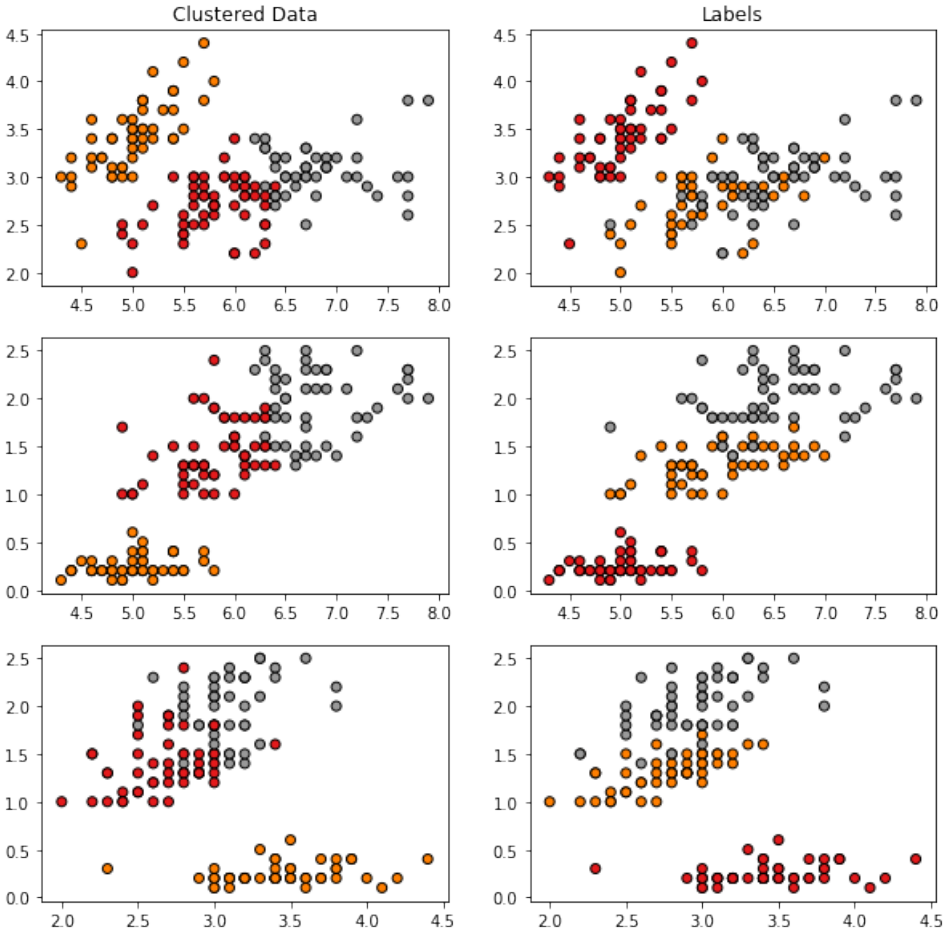
در این بخش ویژگی Sepal Length را حذف و سپس خوشه‌بندی می‌کنیم. در شکل (۱۱) نمودار های سمت چپ مربوط به نتایج خوشه‌بندی و نمودارهای سمت راست مربوط به لیبل‌های واقعی گل‌هاست. صحت خوشه‌بندی تغییر محسوسی نیافته است.



شکل ۱۱: بررسی خوشه‌بندی ـ نمودار سمت راست بر حسب لیبل‌های واقعی و نمودار سمت چپ بر اساس خوشه‌بندی است. در این‌جا ویژگی Sepal Length حذف شده است.

۳.۱۰ حذف Petal Length

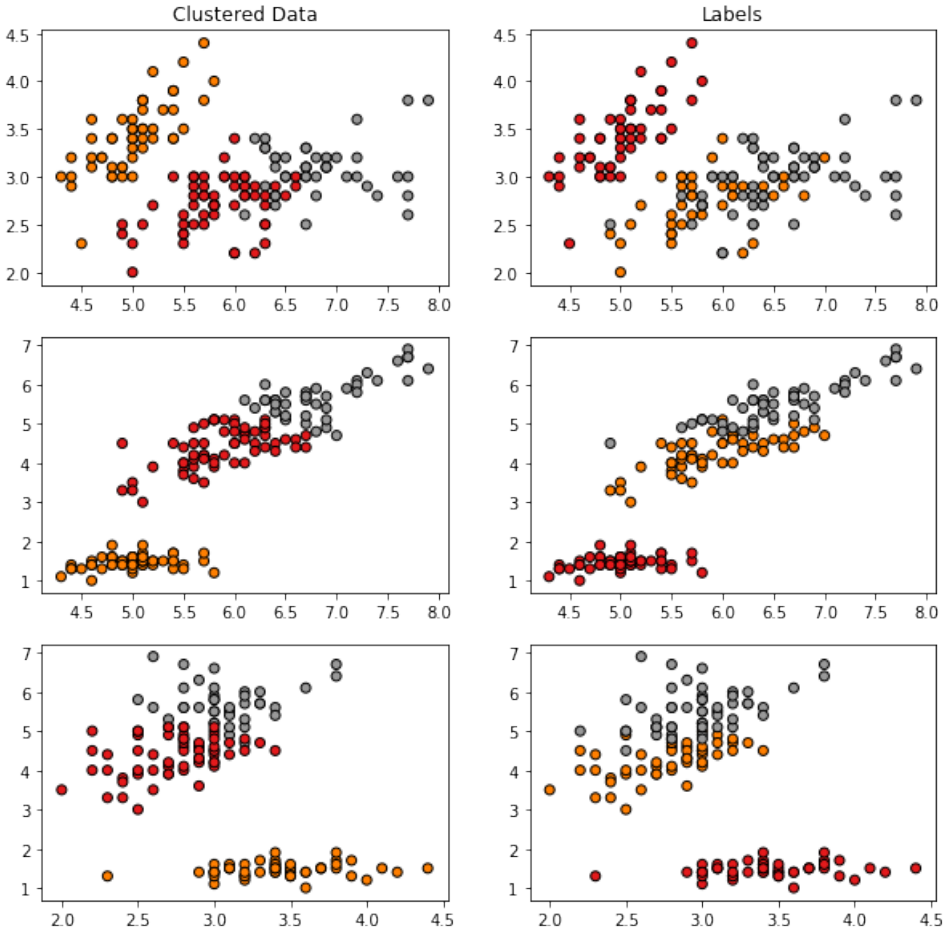
در این بخش ویژگی Petal Length را حذف و سپس خوشه‌بندی می‌کنیم. در شکل (۱۲) نمودار های سمت چپ مربوط به نتایج خوشه‌بندی و نمودارهای سمت راست مربوط به لیبل‌های واقعی گل‌هاست. صحت خوشه‌بندی تغییر محسوسی نیافته است.



شکل ۱۲: بررسی خوشه‌بندی ـ نمودار سمت راست بر حسب لیبل‌های واقعی و نمودار سمت چپ بر اساس خوشه‌بندی است. در این‌جا ویژگی Petal Length حذف شده است.

۴.۱۰ حذف Petal Width

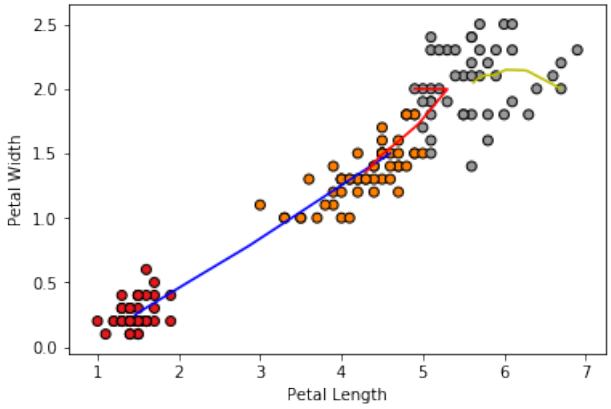
در این بخش ویژگی Petal Width را حذف و سپس خوشه‌بندی می‌کنیم. در شکل (۱۲) نمودار های سمت چپ مربوط به نتایج خوشه‌بندی و نمودارهای سمت راست مربوط به لیبل‌های واقعی گل‌هاست. صحت خوشه‌بندی تغییر محسوسی نیافته است.



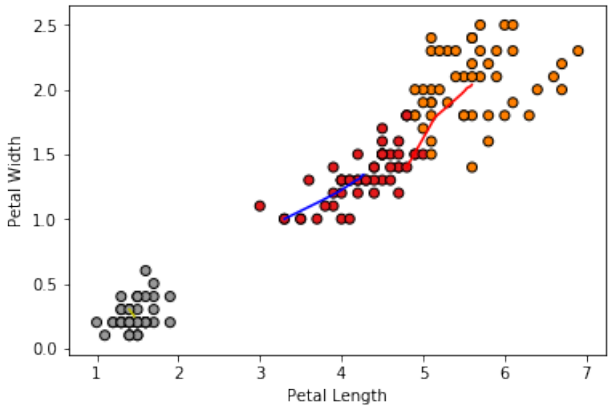
شکل ۱۳: بررسی خوشه‌بندی ـ نمودار سمت راست بر حسب لیبل‌های واقعی و نمودار سمت چپ بر اساس خوشه‌بندی است. در این‌جا ویژگی Petal Width حذف شده است.

۱۱ بررسی تغییرات مراکز خوشه‌ها

در این بخش، با تغییر کد الگوریتم K-Means در هر مرحله، مراکز خوشه‌ها را ذخیره کرده و در انتها مسیر حرکت آن‌ها را در طی تکرارها رسم می‌کنیم. در این‌جا ما تنها از دو ویژگی Petal Width و Petal Length استفاده می‌کنیم. در شکل (۱۴) و (۱۵) خروجی خوشه‌بندی و همچنین مسیر حرکت مراکز خوشه‌ها را مشاهده می‌کنید.



شکل ۱۴: بررسی تغییرات مراکز خوشه‌ها



شکل ۱۵: بررسی تغییرات مراکز خوشه‌ها