

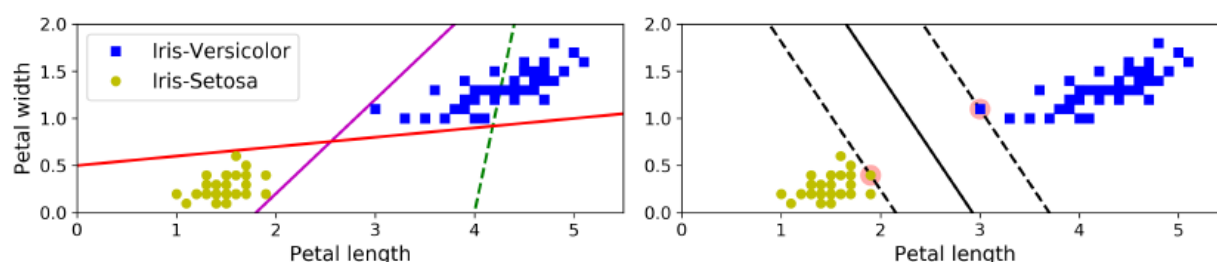
Support Vector Machines

مقدمه :

ماشین های بردار پشتیبان مدل یادگیری ماشین بسیار قدرتمند و همه کاره ای هستند که قادر به طبقه بندی خطی ، غیر خطی ، رگرسیون و حتی تشخیص داده های پرت هستند. SVM ها برای طبقه بندی داده های پیچیده اما کوچک یا متوسط بسیار مناسب هستند.

طبقه بندی خطی SVM :

در طبقه بندی به روش SVM ، بین نمونه های دو کلاس خطی رسم می شود. سپس نمونه جدید اگر در سمت راست خط باشد به عنوان نمونه آن کلاس در نظر گرفته می شود و اگر سمت چپ خط باشد به عنوان نمونه کلاس دیگر در نظر گرفته می شود.



در شکل بالا که طبقه بندی iris را به روش SVM انجام داده است ؛ شکل سمت چپ ، خط رسم شده مناسب نیست زیرا خط رسم شده بسیار به نمونه ها نزدیک است و عملکرد خوبی نخواهد داشت اما شکل سمت راست مناسب است.

هدف SVM پیدا کردن بهترین خط است. برای پیدا کردن بهترین خط ، از هر کلاس یا طبقه یک نمونه را انتخاب می کند. این دو نمونه از دو کلاس متفاوت ، نزدیک ترین نمونه ها به هم ، از دو کلاس باید باشند. سپس از هر نمونه خطی رسم می کند که دو خط با هم دیگر موازی هستند. اصطلاحاً به این خطوط Margin می گویند. ما به دنبال دو خطی خواهیم بود که حداکثر فاصله را با هم داشته باشند. سپس خط جدا کننده یا همان خطی که به دنبالش بود را در وسط و بین این دو خط Margin در نظر می گیرد.

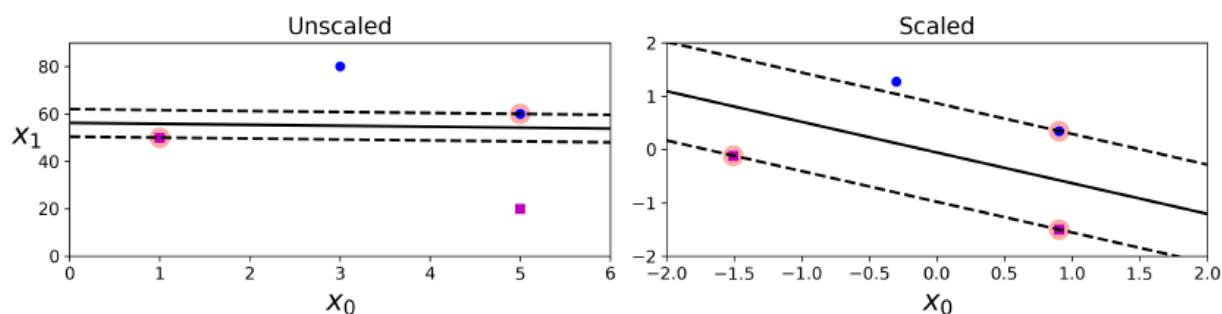
توجه داشته باشید دو خط زمانی موازی هستند که شیب آن ها برابر باشد.

❖ Hard Margin vs Soft Margin :

در مدل SVM اگر دچار مشکل Overfitting شویم یا به عبارت دیگر بخواهیم مدل بر تمامی داده های آموزش fit شود ، در بعضی اوقات ممکن است داده های پرت (Outlier) وجود داشته باشد که این امر باعث می شود در طبقه بندی مدل به خوبی کار نکند. در این حالت اصطلاحاً می گویند Hard Margin رخ داده است. برای آن که مشکل Overfitting رفع شود نیاز است تا داده های پرت را در نظر نگیریم و بدون آن ها مدل را آموزش دهیم که به این حالت Soft Margin می گویند.

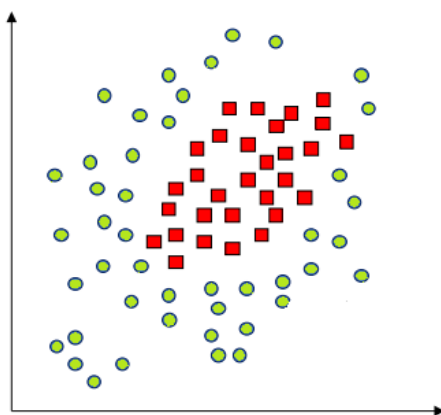
نکته ! SVM به مقیاس ویژگی ها حساس است. در شکل زیر اثر مقیاس بندی ویژگی را در این مدل

می بینیم :

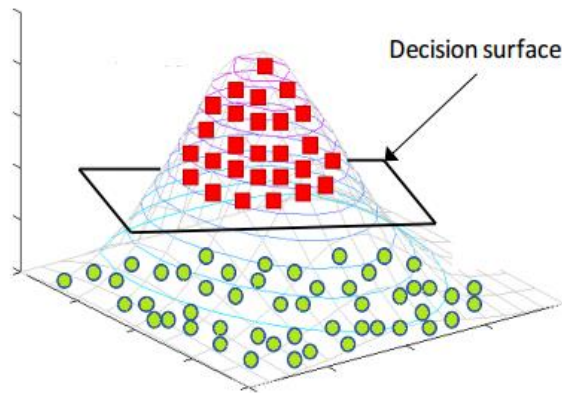


📊 طبقه بندی غیرخطی SVM :

تا به الآن با طبقه بندی خطی SVM آشنا شدیم اما اگر داده ها به صورت خطی از هم تفکیک نشده باشند و نتوان خط راستی بین داده ها برای طبقه بندی آن ها رسم کرد ، چه کار باید کرد. برای مثال اگر پراکندگی داده ها به شکل زیر باشد :



در این حالت دیگر قادر به رسم یک خط برای طبقه بندی نخواهیم بود. برای حل این مشکل می توان از فضای دو بعدی به فضای سه بعدی رفت و در فضای سه بعدی می توان صفحه ای با ویژگی خطی برای طبقه بندی رسم کرد :



بعد سوم با استفاده از دو بعد اصلی ساخته می شود. اصطلاحاً به این صفحات Kernel Trick می گویند. کرنل های متفاوتی برای ایجاد صفحه جدا کننده وجود دارد.

پروژه iris:

می خواهیم با استفاده از دیتاست iris و با استفاده از الگوریتم SVM برنامه ای بنویسیم که ویژگی های مربوط به گل ها را به عنوان داده جدید دریافت کند و آن ها را طبقه بندی کند.
داریم:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import classification_report
```

```
def load_data():
    DataSet = pd.read_csv("iris.csv", header = None,
                          names = ["sepal length",
                                   "sepal width",
                                   "petal length",
                                   "petal width",
                                   "label"])

    data = DataSet.iloc[:, :4]
    label = DataSet.iloc[:, 4]

    return data, label
```

```
data, label = load_data()
```

```
X_train, X_test, y_train, y_test = train_test_split(data, label, test_size=0.3, random_state=30)
```

```
svm_clf = SVC()
```

```
svm_clf.fit(X_train, y_train)
```

```
y_pred = svm_clf.predict(X_test)
```

```
result = classification_report(y_test, y_pred)
```

```
print(result)
```