

## Support Vector Machine

1. Jelaskan cara kerja dari algoritma tersebut! (boleh dalam bentuk *pseudocode* ataupun narasi)

⇒ Pertama, model menerima data latihan dan juga informasi mengenai *learning rate*, jumlah iterasi, dsb. Kemudian model melakukan inisiasi nilai *weights* dan *bias* dengan nilai nol.

Setelah itu, model melakukan proses iterasi sebanyak jumlah iterasi yang telah ditentukan. Pada tiap iterasinya, model akan memperbarui nilai *weight* dan *bias*. Untuk algoritma pembaruan nilai *weight* dan *bias* digunakan juga *soft margin* agar model tetap dapat berjalan meskipun data tidak *linearly separable*.

```
margin = y_[idx] * (np.dot(x_i, self.w) - self.b)
if(margin >= 1):
    self.w -= self.lr * (self.C * self.w)
else:
    self.w -= self.lr * (self.C * self.w - np.dot(x_i, y_[idx]))
    self.b -= self.lr * y_[idx]
```

Setelah proses pelatihan berakhir, nilai *weight* dan *bias* yang terakhir dihasilkan akan digunakan untuk melakukan prediksi. Namun, karena hasil akhir dari prediksi dapat bernilai negatif (sedangkan klasifikasi hanya 0 atau 1), maka dilakukan penyesuaian lagi dimana yang memiliki nilai negatif akan diset menjadi nol nilainya.

2. Bandingkan hasil evaluasi model from scratch dan *library*, bagaimana hasil perbandingannya? Jika ada perbedaan, jelaskan alasannya!

Akurasi [holdout 80-20]	
<i>Scratch</i>	<i>Library</i>
0.462	0.452
Akurasi [k-fold 5]	

<b><i>Scratch</i></b>	<b><i>Library</i></b>
0.609	0.531
<b>Waktu Eksekusi</b>	
<b><i>Scratch</i></b>	<b><i>Library</i></b>
2 detik	0.018 detik

3. Jelaskan *improvement* apa saja yang bisa Anda lakukan untuk mencapai hasil yang lebih baik dibandingkan dengan hasil yang Anda punya saat ini! *Improvement* yang dimaksud tidak terbatas pada bagaimana algoritma diimplementasikan, namun juga mencakup tahap sebelum *modeling and validation*.

⇒ Preprocessing : *scaling* data akan membantu meningkatkan kinerja SVM. Untuk melakukan *scaling* data dapat menggunakan StandardScaler ataupun RobustScaler. StandardScaler digunakan ketika data yang dimiliki terdistribusi secara normal, sedangkan RobustScaler dapat dilakukan pada data yang tidak terdistribusi normal karena menggunakan *interquartile range*, sedangkan StandardScaler menggunakan rata-rata dan standar deviasi.

⇒ Hyperparameter tuning : memilih hyperparameter yang tepat mulai dari jenis kernel (pada model yang dibuat baru hanya mengimplementasikan kernel linear saja) yang digunakan ataupun nilai parameter C dapat mempengaruhi hasil yang dihasilkan.