Hoofdstuk 1

Het model

1.1 Bias

In machine learning verwijst bias naar de systematische fout die door een model wordt geïntroduceerd wanneer het de onderliggende patronen in de trainingsgegevens consequent verkeerd weergeeft. Dit kan leiden tot onnauwkeurige voorspellingen en onbetrouwbare prestaties op nieuwe datasets, zoals de validatie- en testdatasets.

Wanneer we bias bespreken in de context van classificatie, verwijzen we vaak naar de neiging van het model om bepaalde klassen boven andere te bevoordelen. Er zijn dan twee belangrijke soorten bias die we willen toelichten.

1.1.1 *Algorithmic bias*

Algorithmic bias doet zich voor wanneer het algoritme zelf is getraind op een manier die bepaalde uitkomsten of groepen bevoordeelt. Het wordt bepaald door de gegevens die worden gebruikt voor training, en de keuze van het type model. Als een classificatiemodel bijvoorbeeld wordt getraind op vertekende gegevens die overwegend één groep vertegenwoordigen, kan het model leren nauwkeuriger te zijn voor die groep, terwijl het slecht presteert voor andere groepen.

Als we ons SVM model bijvoorbeeld bijna uitsluitend trainen op gevallen waar de tumor kwaadaardig is, zal het hoogstwaarschijnlijk niet goed presteren voor nieuwe datapunten waar de tumor goedaardig is. Het model is namelijk niet getraind op tumoren van die aard en zal dus geen nauwkeurige predicties kunnen doen.

1.1.2 *Sampling bias*

Sampling bias treedt op wanneer de trainingsgegevens niet representatief zijn voor de populatie waarnaar ze willen generaliseren. Dit kan gebeuren als bepaalde groepen in de populatie onder- of oververtegenwoordigd zijn in het trainingspakket. Als we ons SVM model bijvoorbeeld enkel trainen op tumoren van vrouwelijke borstkankerpatiënten, is de kans dat het model slecht presteert voor een nieuwe dataset van mannelijke patiënten vrij groot.

1.2 Variantie

De variantie is de maat voor de gevoeligheid van het model. Het geeft inzicht over de flexibiliteit van een model, met name hoe nauwkeurig de voorspellingen zijn bij verschillende datasets.

Bij een hoge variantie is het model sterk aangepast aan de trainingsdata. Dit betekent dat zelfs kleine veranderingen in de trainingsdata een grote invloed zullen hebben op het model, wat resulteert in een fenomeen genaamd *overfitting*. In het geval van tumorclassificatie zou een hoge variantie betekenen dat het model zeer nauwkeurig kan voorspellen of een tumor goedaardig is bij de trainingsdata. Maar doordat het model te sterk is afgestemd op de trainingsdata, zal het moeite hebben met generaliseren en levert dit geen precieze voorspellingen op voor de andere datasets, met name de validatie- en testdatasets.

Aan de andere kant kan een model ook een te lage variantie hebben. Dit betekent dan dat het model te veel generaliseert en weinig verandert indien de trainingsdata verandert. Een te lage variantie leidt echter tot *underfitting*. Hierdoor levert het ook geen betrouwbare voorspellingen op. Toegepast op tumorclassificatie zal een te lage variantie leiden tot een te sterk gegeneraliseerd model, waarbij het model niet goed presteert op de 3 verschillende datasets.

1.3 Flexibiliteit van het model bepalen

Hoe bepalen we nu de ideale flexibiliteit van ons model? Met andere woorden: hoe voorkomen we zowel *overfitting*, waarbij het model de trainingsdata te nauwgezet probeert te volgen, als *underfitting*, waarbij er sprake is van overgeneralisatie?

1.3.1 Metaparameters

Machine learning modellen hebben over het algemeen een of meerdere *metaparameters* die de flexibiliteit van het model beïnvloeden. Het model zal zich meer of minder aanpassen naar gelang de waarden van deze metaparameter(s). Zoals we later zullen zien in hoofdstuk **??**, is een zekere λ de metaparameter die bij Support Vector Machines van belang is. Deze parameter wordt voor SVM ook wel de *regularisatieparameter* genoemd. Deze λ zal uiteindelijk de breedte van de marge - en dus de variabiliteit van het model - beïnvloeden (zie later).

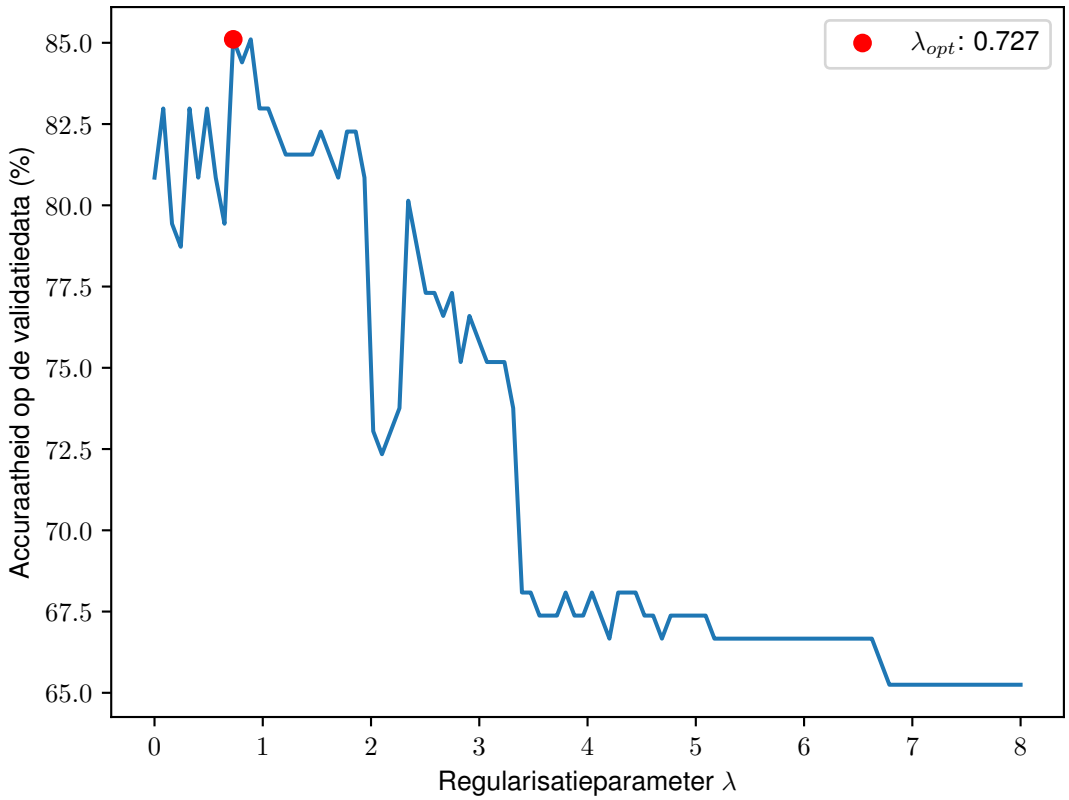
1.3.2 Cross-validation

Om de optimale metaparameters te bepalen, gebruiken we het principe van *cross-validation*. Hiervoor splitsen we eerst de data die we voor handen hebben op in 3 delen: 50% trainingsdata, 25% validatiedata, 25% testdata.

De eerste stap is het trainen van ons model op de trainingsdata met een bepaalde waarde voor de metaparameter(s). Hierna zullen we de accuraatheid van dit model voor die bepaalde metaparameter(s) testen op de validatiedataset. Dit proces herhalen we voor - liefst zo veel mogelijk - verschillende waardes van de metaparameter(s), tot we de grootste accuraatheid van de voorspelling voor de trainingsdata hebben verkregen. De metaparameter(s) die voor de grootste accuraatheid zorgen op de trainingsdata, zullen we dan beschouwen als 'optimaal'.

We kunnen de accuraatheid van het getrainde model t.o.v. de trainingsdata uitzetten in functie van een metaparameter in een grafiek, zoals te zien is in figuur 1.1. De waarde van de metaparameter waarvoor de grafiek die de accuraatheid weergeeft een maximum bereikt, zal de optimale waarde voor deze metaparameter zijn.

Tot slot gebruiken we onze testset om de algemene prestatie van ons model te beoordelen. Het uiteindelijke doel is om een zo goed mogelijke voorspelling te kunnen maken, gegeven een nieuwe set tumoren. We willen dus op basis van de 30 kenmerken de aard van nog niet eerder geziene tumoren voorspellen.



Figuur 1.1: De accuraatheid van het model t.o.v. de trainingsdata, uitgezet in functie van de metaparameter λ voor ons SVM-model.