Literatuuronderzoek

Table of Contents

Machine learning:

Er zijn verschillende soorten machine learning die worden gebruikt voor verschillende doeleinden. Hier zijn enkele van de belangrijkste categorieën:

Supervised learning: Dit is de meest voorkomende vorm van machine learning waarbij een model wordt getraind op een dataset met bekende output. Hieronder vallen o.a. linear regression, logistic regression, decision tree, en support vector machine.

Unsupervised learning: Dit is een vorm van machine learning waarbij een model wordt getraind op een dataset zonder bekende output. Hieronder vallen o.a. k-means, hierarchical clustering en association rule learning.

Reinforcement learning: Dit is een vorm van machine learning waarbij een agent leert door middel van interactie met zijn omgeving. Hieronder vallen Q-Learning, SARSA, en DQN.

Deep learning: Dit is een subcategorie van machine learning die gebruik maakt van diep neural networks. Hieronder vallen Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), en Generative Adversarial Networks (GAN).

Semi-supervised learning: Dit is een vorm van machine learning waarbij een model wordt getraind op een dataset met een combinatie van bekende en onbekende output.

Transfer learning: Dit is een vorm van machine learning waarbij een model wordt getraind op een dataset en deze kennis wordt toegepast op een ander model of een andere dataset.

De keuze van welk type machine learning te gebruiken hangt af van de specifieke toepassing en de beschikbaarheid van data.

Supervised learning:
Supervised learning is een type machine learning waarbij een algoritme getraind wordt met behulp van gegevens waarvan de uitkomst (de "label") al bekend is.
Het doel van het trainen is om het algoritme in staat te stellen nieuwe, onbekende gegevens te classificeren of te voorspellen op basis van wat het heeft geleerd uit de getrainde gegevens.

Unsupervised learning:
Unsupervised learning is een type machine learning waarbij geen uitkomst of "label" bekend is. Het algoritme moet zelf structuren of patronen in de gegevens ontdekken. Er zijn twee hoofdtypen unsupervised learning: clustering en dimensie-reductie.
Clustering is gericht op het vinden van groepen of clusters in de gegevens die vergelijkbare kenmerken hebben. Hierdoor kunnen bijvoorbeeld onbekende groepen of categorieën in de data worden ontdekt. Bekende methoden zijn k-means, hierarchical clustering en density-based clustering.
Dimensie-reductie is gericht op het verminderen van de hoeveelheid features in de gegevens zonder informatie te verliezen. Hierdoor kunnen bijvoorbeeld complexe gegevens worden vereenvoudigd of visueel weergegeven. Bekende methoden zijn principal component analysis (PCA), t-SNE, en autoencoders.
Unsupervised learning wordt vaak gebruikt voor exploratie van gegevens, bijvoorbeeld om onbekende groepen of patronen in de gegevens te ontdekken, of om gegevens voor te bereiden voor verdere analyse of classificatie.

Reinforcement learning (RL):
Reinforcement learning (RL) is een type machine learning waarbij een agent leert door middel van trial-and-error, door de interactie met zijn omgeving. Het model moet een reeks van acties kiezen om een specifiek doel te bereiken, en ontvangt feedback in de vorm van beloningen of straffen.
In RL wordt gebruik gemaakt van een beloningsfunctie die aangeeft hoe goed de huidige staat van de agent is in vergelijking met het doel. De agent leert door te proberen verschillende acties uit te voeren en te evalueren welke acties leiden tot de grootste beloning.
Er zijn twee hoofdtypen RL: value-based en policy-based. Value-based methoden proberen de verwachte beloning van een staat of actie te schatten, terwijl policy-based methoden direct een beleid (actie) leren voor elke staat.
RL wordt vaak gebruikt in gebieden zoals gaming, robots, en financieel beleid. Het wordt ook gebruikt in problemen waar het onmogelijk is om de complete set van gegevens of de juiste oorzaak-gevolg relatie te kennen.

De verschillende modellen:

Lineaire regressie:
Lineaire regressie is een statistisch model dat gebruikt wordt om de relatie tussen één of meerdere onafhankelijke variabelen (x) en één afhankelijke variabele (y) te beschrijven.
Het model gaat uit van een lineaire relatie tussen x en y, oftewel een rechte lijn. Deze rechte lijn wordt bepaald door de parameters van het model, waarvan de waarden worden aangepast tijdens het trainingsproces zodat deze zo goed mogelijk de werkelijkheid benaderen.
Het doel van lineaire regressie is om een formule te vinden die zo goed mogelijk de waarnemingen van de afhankelijke variabele (y) kan voorspellen op basis van de waarnemingen van de onafhankelijke variabele(n) (x).
Het wordt vaak gebruikt om een voorspelling te doen van de waarde van de afhankelijke variabele op basis van de waarde van de onafhankelijke variabele(n).

Logistische regressie:
Logistische regressie is een statistisch model dat gebruikt wordt voor voorspelling van binair gecodeerde waarden (bijvoorbeeld true/false of 1/0). Het model gaat uit van een logistische relatie tussen de invloed van één of meerdere onafhankelijke variabelen (x) en de kans op een bepaalde uitkomst (y). Deze relatie wordt beschreven door een logistische functie, waarvan de parameters worden aangepast tijdens het trainingsproces zodat deze zo goed mogelijk de werkelijkheid benaderen.
Het doel van logistische regressie is het bepalen van de kans op een bepaalde uitkomst op basis van de waarde van de onafhankelijke variabelen. Hierdoor is het een classificatie model gebaseerd op kansberekening.
Het wordt vaak gebruikt om te bepalen of een bepaald patroon in gegevens een specifieke uitkomst voorspelt, bijvoorbeeld kans op ziekte, of kans op een bepaald gedrag.

Decision tree:
Decision tree is een algoritme voor besluitvorming en classificatie. Het is een grafisch model waarbij de data worden gepresenteerd in de vorm van een boomstructuur. Elke knoop in de boom vertegenwoordigt een beslissing of een voorwaarde, en elke tak uit een knoop leidt naar een volgende knoop of een eindresultaat (leaf).
Het doel van decision tree is om een set van voorwaarden te creëren die een zo hoog mogelijke classificatie-accuraatheid geven. Het algoritme begint met het identificeren van de meest relevante feature (onderdeel van de gegevens) om een eerste splitsing te maken. Vervolgens wordt deze splitsing herhaald voor de subgroepen die zijn ontstaan totdat elke groep bestaat uit gegevens met dezelfde label.
Decision tree wordt vaak gebruikt voor het maken van beslissingen en het oplossen van problemen waarbij een groot aantal voorwaarden of criteria moet worden overwogen. Het is een visueel sterk model dat het proces van besluitvorming verduidelijkt, en kan ook gemakkelijk worden geëxporteerd naar code voor automatisering.

Random forest:
Random forest is een algoritme voor classificatie en regressie. Het is een verzameling van decision trees, waarbij elke boom is getraind op een willekeurige subset van de gegevens en een willekeurige subset van de features. De uiteindelijke voorspelling van een random forest wordt bepaald door een combinatie van voorspellingen van elke boom in de verzameling.
Een van de voordelen van random forest is dat het de kans op overfitting verkleint. Overfitting treedt op wanneer een model te specifiek is afgestemd op de trainingsgegevens en presteert slecht op nieuwe, onbekende gegevens. Dit gebeurt vaak bij decision trees wanneer er te veel splitsingen

zijn gemaakt. Door het gebruik van een groot aantal beslissingsbomen die elk op een willekeurige subset van de gegevens zijn getraind, wordt de kans op overfitting verkleind.
Daarnaast, door het gebruik van een groot aantal bomen, random forest kan ook beter omgaan met 'ruis' in de data, of met datapunten die niet goed in een specifieke categorie passen. Een ander voordeel is dat het de feature importances kan bepalen, dat wil zeggen welke features het meest van invloed zijn op de uitkomst.
Random forest wordt vaak gebruikt voor complexe problemen waarbij veel features van invloed zijn, of waarbij de data ruis bevat.

Neural networks:
Neural networks (NN) zijn een soort van machine learning algoritmen geïnspireerd door de structuren en functies van neuronen in de hersenen. Het zijn complexe modellen die bestaan uit meerdere lagen van zogenoemde neuronen, die verbonden zijn met elkaar.
Deze neuronen ontvangen informatie via hun input, verwerken deze informatie en sturen een output door naar de volgende laag neuronen. Zo verwerkt elke laag informatie op een andere manier, en de gehele NN leert zo om complexe patronen in de data te herkennen.
NN worden vaak gebruikt voor zeer complexe problemen zoals beeld- en taalherkenning, en voor het analyseren van grote hoeveelheden gegevens. Er zijn verschillende soorten NN zoals perceptron, feedforward NN, recurrent NN, deep learning NN, die elk geschikt zijn voor specifieke soorten problemen.
Het trainen van een NN gebeurt door middel van backpropagation, waarbij de fouten die gemaakt worden tijdens het voorspellen van de uitkomst, gebruikt worden om de parameters van de NN aan te passen, zodat deze fouten in de toekomst verminderen.

SVM (Support Vector Machine):
SVM (Support Vector Machine) is een algoritme voor supervised learning, dat gebruikt wordt voor classificatie en regressie. Het model is gebaseerd op de idee om de gegevens te scheiden in twee groepen door middel van een hyperplane, een rechte die zo is gekozen dat de afstand tussen de hyperplane en de dichtstbijzijnde gegevenspunten van beide groepen zo groot mogelijk is. Deze dichtstbijzijnde gegevenspunten heten support vectors.
SVM is in staat om niet-lineaire beslissingsgrenzen te creëren door het gebruik van kernel-functies. Deze functies zetten de oorspronkelijke gegevens om naar een hogere dimensie waar een lineaire beslissingsgrens kan worden gevonden.
SVM wordt vaak gebruikt voor kleine datasets of datasets met veel features, omdat het efficiënt is in het gebruik van geheugen en tijd. Het is ook effectief in het oplossen van problemen met veel ruis of overlappende classes.

K nearest neighbors (k-NN):
K nearest neighbors (k-NN) is een algoritme voor supervised learning, dat gebruikt wordt voor classificatie en regressie. Het is een instance-based learning algoritme, wat betekent dat het geen expliciete model maakt van de gegevens, maar de beslissingen baseert op de gegevens zelf.
Het algoritme werkt door een gegeven dat moet worden gecategoriseerd of geprediceerd, te vergelijken met de k "dichtstbijzijnde" gegevens in de training set, waarbij "dichtstbijzijnde" wordt bepaald door een bepaalde afstandsmeting. De categorie of waarde van de meerderheid van de k "dichtstbijzijnde" gegevens wordt toegekend aan het te classificeren of te voorspellen gegeven.
De grootte van k is een belangrijke parameter in het algoritme en bepaalt hoeveel invloed elk individueel punt heeft op de uiteindelijke beslissing. Een kleine waarde van k kan leiden tot overfitting, terwijl een grotere waarde van k kan leiden tot onderfitting.
k-NN wordt vaak gebruikt voor gegevens met weinig features en is een eenvoudig te begrijpen en te implementeren algoritme. Het is ook geschikt voor continu geparametriseerde gegevens en kan goed omgaan met gegevens met veel ruis.

Q-learning:
Q-learning is een type machine learning-algoritme dat wordt gebruikt voor het oplossen van problemen met een beperkt aantal staat-actieparen. Het is een soort van reïnforcements learning waar Q-value voor elke staat-actiepaar wordt bijgehouden. De Q-value geeft aan hoe waarschijnlijk het is dat een specifieke actie leidt tot een optimale beloning in de toekomst. Tijdens het trainingsproces worden de Q-waarden aangepast op basis van de beloning die wordt ontvangen na het uitvoeren van een actie. Op deze manier kan het algoritme leren welke acties het meest waarschijnlijk leiden tot een optimale oplossing.

Sarsa (state-action-reward-state-action):
Sarsa (state-action-reward-state-action) is een soort van Q-learning-algoritme dat wordt gebruikt voor het oplossen van problemen met een beperkt aantal staat-actieparen. Net als Q-learning houdt het algoritme Q-waarden bij voor elke staat-actiepaar, maar in plaats van te kiezen voor de actie met de hoogste Q-waarde, kiest Sarsa voor de volgende actie op basis van de huidige staat en de huidige Q-waarden. Hierdoor kan Sarsa rekening houden met de context waarin een actie wordt uitgevoerd, wat kan leiden tot betere prestaties in sommige gevallen.

Deep Q-Learning (DQL):
Deep Q-Learning (DQL) is een variatie van Q-learning die gebruikmaakt van neural networks om de Q-waarden te schatten in plaats van een tabel te gebruiken. Dit maakt het mogelijk om met veel meer staat-actieparen te werken.
De Q-waarden worden nu geschat door een deep neural network met de huidige staat als input en de Q-waarden voor alle mogelijke acties als output. Tijdens het trainingsproces wordt het netwerk getraind om de Q-waarden voor elke staat-actiepaar zo nauwkeurig mogelijk te schatten. Dit wordt gedaan door het netwerk te vergelijken met de daadwerkelijke Q-waarden die worden verkregen uit het uitvoeren van acties in de omgeving.
Het gebruik van neural networks maakt DQL veel flexibeler en in staat om met veel meer staat-actieparen te werken dan traditionele Q-learning algoritmen.

Convolutional Neural Network (CNN):
Een Convolutional Neural Network (CNN) is een soort van deep learning-architectuur die specifiek is ontworpen voor het analyseren van afbeeldingen. Een CNN bestaat uit een aantal lagen, waaronder convolutionele lagen, pooling lagen en fully connected lagen.
De convolutionele lagen zijn verantwoordelijk voor het extraheren van kenmerken uit de afbeelding, zoals lijnen, hoeken en patronen. Dit gebeurt door het toepassen van filters op de afbeelding, die kleine delen van de afbeelding scannen en kenmerken extraheren.
Pooling lagen worden gebruikt om de grootte van de afbeelding te verkleinen en de kenmerken te versterken die door de convolutionele lagen zijn gedetecteerd.
De fully connected lagen zijn verantwoordelijk voor het classificeren van de afbeelding op basis van de kenmerken die zijn gedetecteerd door de convolutionele en pooling lagen.
CNNs zijn zeer effectief in het oplossen van allerlei beeldherkenning taken zoals object detection, image classification, en semantic segmentation.

Deep Q-Network (DQN):
Deep Q-Network (DQN) is een specifiek type van Deep Q-Learning (DQL) model dat gebruik maakt van een neural network om de Q-waarden te schatten. Dit netwerk wordt ook wel een Q-netwerk genoemd. DQN introduceerde een aantal verbeteringen op de traditionele Q-learning, waaronder het gebruik van een experience replay buffer en een target network.

De experience replay buffer wordt gebruikt om eerdere ervaringen op te slaan en later te gebruiken voor het trainen van het Q-netwerk. Dit helpt om de correlatie tussen opeenvolgende ervaringen te verminderen en het algoritme te stabiliseren.
Het target network is een tweede kopie van het Q-netwerk die wordt gebruikt voor het berekenen van de verwachte Q-waarde in plaats van de Q-waarde die wordt geschat door het Q-netwerk zelf. Dit helpt om het probleem van oscillerende Q-waarden te verminderen.
DQN wordt vaak gebruikt voor problemen met een beperkt aantal staat-actieparen, zoals Atari-spellen, waarbij het algoritme in staat is om een optimale spelstrategie te leren door middel van trial and error.

Overig:

Overfitting en underfitting zijn twee veel voorkomende problemen bij het trainen van machine learning-modellen.

Overfitting gebeurt wanneer een model te veel aanpast aan de training data en daardoor niet goed presteert op nieuwe, onbekende data. Dit komt omdat het model te veel details van de training data heeft geleerd, waardoor het niet algemeen genoeg is. Er zijn verschillende manieren om overfitting te voorkomen, zoals:

Het verkleinen van de complexiteit van het model (bijvoorbeeld door het gebruik van minder beslissingsbomen in een random forest)
Het verkleinen van de hoeveelheid features die worden gebruikt.
Het toepassen van regulatie zoals L1 of L2 regulatie
Het toevoegen van meer training data.
Underfitting gebeurt wanneer een model te weinig aanpast aan de training data en daardoor slecht presteert op zowel de training data als op nieuwe, onbekende data. Dit komt omdat het model te weinig informatie heeft geleerd uit de training data. Er zijn verschillende manieren om underfitting te voorkomen, zoals:

Het vergroten van de complexiteit van het model
Het toevoegen van meer features
Het toepassen van ensemble methoden
Het toevoegen van meer training data.
Het is belangrijk om te weten dat het voorkomen van overfitting en underfitting vaak een trade-off is en dat er een balans moet worden gevonden tussen de complexiteit van het model en de hoeveelheid beschikbare data.

Een confusion matrix, ook wel foutenmatrix genoemd, is een tool die wordt gebruikt om de prestaties van een classificatiemodel te evalueren. Het geeft inzicht in hoe vaak een model de juiste voorspelling doet en hoe vaak het een foutieve voorspelling doet.

Een confusion matrix bestaat uit vier verschillende types voorspellingen: true positives (TP), true negatives (TN), false positives (FP) en false negatives (FN).

True positives (TP) zijn voorspellingen waarbij het model een positieve uitkomst voorspelt en de daadwerkelijke uitkomst ook positief is.
True negatives (TN) zijn voorspellingen waarbij het model een negatieve uitkomst voorspelt en de daadwerkelijke uitkomst ook negatief is.
False positives (FP) zijn voorspellingen waarbij het model een positieve uitkomst voorspelt, maar de daadwerkelijke uitkomst negatief is.
False negatives (FN) zijn voorspellingen waarbij het model een negatieve uitkomst voorspelt, maar de daadwerkelijke uitkomst positief is.
De confusion matrix wordt vaak gepresenteerd als een tabel waarbij de verticale as de daadwerkelijke uitkomst vertegenwoordigt en de horizontale as de voorspelde uitkomst.

Er zijn ook verschillende metrices die kunnen worden berekend aan de hand van een confusion matrix, zoals accuracy, precision, recall, f1-score en ROC AUC.

De accuracy geeft aan hoe vaak het model de juiste voorspelling doet, precision geeft aan hoe vaak een positieve voorspelling juist is, recall geeft aan hoe vaak een daadwerkelijke positieve uitkomst juist wordt voorspeld, f1-score is de harmonic mean van precision en recall en ROC AUC is de area under the curve of the receiver operating characteristic.

Bibiliography

1. Kurian, J. (2020, September 14). Visualizing Decision Trees in Jupyter Notebook with Python and Graphviz. Retrieved January 16, 2023, from https://towardsdatascience.com/visualizing-decision-trees-in-jupyter-notebook-with-python-and-graphviz-78703230a7b1
2. GeeksforGeeks. (n.d.). SARSA Reinforcement Learning. Retrieved January 16, 2023, from https://www.geeksforgeeks.org/sarsa-reinforcement-learning/
3. Kurian, J. (2020, September 14). Multiple Linear Regression Model Using Python: Machine Learning. Retrieved January 16, 2023, from https://towardsdatascience.com/multiple-linear-regression-model-using-python-machine-learning-d00c78f1172a
4. Subconscious Musings. (2020, December 9). Machine Learning Algorithm Use. Retrieved January 16, 2023, from https://blogs.sas.com/content/subconsciousmusings/2020/12/09/machine-learning-algorithm-use/
5. Kurian, J. (2020, September 14). All Machine Learning Models Explained in 6 Minutes. Retrieved January 16, 2023, from https://towardsdatascience.com/all-machine-learning-models-explained-in-6-minutes-9fe30ff6776a
6. JavaTpoint. (n.d.). Machine Learning Models. Retrieved January 16, 2023, from https://www.javatpoint.com/machine-learning-models
7. DataCamp. (n.d.). Decision Tree Classification in Python. Retrieved January 16, 2023, from https://www.datacamp.com/tutorial/decision-tree-classification-python
8. GeeksforGeeks. (n.d.). Decision Tree Implementation in Python. Retrieved January 16, 2023, from https://www.geeksforgeeks.org/decision-tree-implementation-python/
9. Simplilearn. (n.d.). Decision Tree in Python. Retrieved January 16, 2023, from https://www.simplilearn.com/tutorials/machine-learning-tutorial/decision-tree-in-python
10. Monkeylearn. (n.d.). Classification Algorithms. Retrieved January 16, 2023, from https://monkeylearn.com/blog/classification-algorithms/