**Fundamentals of machine learning**

**Introduktion till Machine Learning**

Machine Learning (ML) handlar om att utveckla algoritmer som kan identifiera mönster i data och använda dessa för att göra förutsägelser eller beslut utan att explicit ha blivit programmerade för uppgiften. Istället för att en utvecklare definierar regler och logik manuellt, använder en ML-modell data för att lära sig sambanden själv.

Den grundläggande processen för att skapa en ML-modell innefattar tre steg: **träning**, **validering** och **inferencing**. Under träningen matas modellen med **training data**, där både input och förväntade output-värden finns tillgängliga. Genom att justera sina parametrar hittar modellen mönster i data. Därefter används valideringsdata för att testa modellens prestanda och finjustera den innan den används för inferencing, det vill säga för att göra förutsägelser på nya data.

En stor utmaning är att säkerställa att modellen inte **overfittar**, det vill säga lär sig för mycket detaljer från träningsdatan och presterar dåligt på ny data. För att förhindra detta används tekniker som **regularization** och **cross-validation**.

**Olika typer av Machine Learning**

Machine Learning delas in i tre huvudkategorier beroende på hur algoritmen tränas och vilken typ av data den hanterar:

1. **Supervised Learning** – Modellen tränas med märkta data (**labeled data**) där både indata och korrekt utdata är kända. Exempel är:
   * **Regression**, där modellen förutspår kontinuerliga värden.
   * **Classification**, där modellen förutspår vilken kategori en observation tillhör.
2. **Unsupervised Learning** – Modellen arbetar med omärkta data (**unlabeled data**) och hittar själv strukturer eller mönster. Det används exempelvis inom **clustering**, där liknande datapunkter grupperas ihop.
3. **Reinforcement Learning** – Modellen lär sig genom att interagera med en miljö och får **belöningar** eller **straff** baserat på sina handlingar. Det används ofta inom **robotik** och **spelteori**.

**Regression: Förutsäga kontinuerliga värden**

Regression är en typ av supervised learning som används för att förutsäga numeriska värden baserat på historiska data. Den vanligaste metoden är **Linear Regression**, där en rak linje skapas för att bäst passa datan. Formeln för en enkel linjär regression är:

y=mx+by = mx + b

där yy är det förutspådda värdet, xx är input, mm är lutningen (koefficienten), och bb är interceptet. Modellen lär sig de optimala värdena för mm och bb genom en process som kallas **gradient descent**, där den iterativt justerar parametrarna för att minimera felmarginalen.

**Evaluation Metrics** som används för att bedöma en regressionsmodell inkluderar:

* **Mean Squared Error (MSE)**
* **Root Mean Squared Error (RMSE)**
* **R-squared (R²)**

Om modellen visar dåliga prediktioner kan det bero på **underfitting** (modellen är för enkel) eller **overfitting** (modellen är för komplex och anpassad till träningsdatan).

**Klassificering: Förutsäga kategoriska etiketter**

**Binary Classification** är en typ av supervised learning där modellen förutspår en av två möjliga kategorier. Ett exempel är en modell som avgör om ett e-postmeddelande är spam eller inte. **Logistic Regression** är en populär algoritm för binär klassificering, trots sitt namn är det inte en regressionsmetod utan en klassificeringsmetod. Den använder **sigmoid-funktionen** för att omvandla en kontinuerlig variabel till en sannolikhet mellan 0 och 1.

Formeln för **sigmoid-funktionen** är:

σ(x)=11+e−x\sigma(x) = \frac{1}{1 + e^{-x}}

När sannolikheten överstiger en viss **threshold**, till exempel 0.5, klassificeras datapunkten som tillhörande en viss kategori.

För att utvärdera en binär klassificeringsmodell används **confusion matrix**, vilket visar:

* **True Positives (TP)**
* **False Positives (FP)**
* **True Negatives (TN)**
* **False Negatives (FN)**
* Vanliga metrik för att bedöma prestanda är **Accuracy**, **Precision**, **Recall**, och **F1-score**.  
    
  **Accuracy** är det enklaste måttet och beräknas som andelen korrekt klassificerade fall i förhållande till det totala antalet observationer:

Accuracy=TP+TNTP+TN+FP+FNAccuracy = \frac{TP + TN}{TP + TN + FP + FN}

Dock kan accuracy vara missvisande om datasetet är **obalanserat** (t.ex. vid sjukdomsdiagnoser där 95% av fallen är friska, en modell som alltid gissar "frisk" har 95% accuracy men är värdelös).

* **Precision** mäter andelen verkliga positiva av alla som modellen klassificerat som positiva:

Precision=TPTP+FPPrecision = \frac{TP}{TP + FP}

Viktigt när **falska positiva** är kostsamma, t.ex. vid spamfilter (du vill inte att viktiga e-postmeddelanden hamnar i skräpposten).

* **Recall (Sensitivity)** mäter hur många av de faktiska positiva som modellen korrekt identifierar:

Recall=TPTP+FNRecall = \frac{TP}{TP + FN}

Viktigt när **falska negativa** är farliga, t.ex. vid cancerdiagnos (det är värre att missa en sjuk patient än att felaktigt markera en frisk).

* **F1-score** är det harmoniska medelvärdet av precision och recall:

F1=2×Precision×RecallPrecision+RecallF1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}

F1-score balanserar precision och recall och är särskilt användbart vid obalanserade dataset.

Låt mig veta om du vill ha någon ytterligare förklaring! 😊

Vid **Multiclass Classification** hanterar modellen fler än två klasser. Två vanliga strategier är:

* **One-vs-Rest (OvR)** – En binär klassificeringsmodell skapas för varje klass.
* **Multinomial Classification** – En modell skapar en sannolikhetsfördelning över alla klasser.

Populära algoritmer inkluderar **Decision Trees**, **Random Forest**, och **Support Vector Machines (SVM)**.

**Clustering: Gruppering av data**

Clustering är en form av unsupervised learning där modellen grupperar datapunkter baserat på likheter. Ett vanligt tillämpningsområde är kundsegmentering, där företag grupperar kunder med liknande köpbeteenden.

**K-Means Clustering** är en populär algoritm där modellen försöker skapa **K** kluster genom att placera **centroids** och iterativt justera dem baserat på de datapunkter som tillhör varje kluster. Algoritmen stoppar när inga datapunkter längre byter kluster.

För att bedöma kvaliteten på klustring används metoder som **Silhouette Score** och **Elbow Method**.

**Deep Learning: Neurala nätverk**

Deep Learning är en avancerad gren av machine learning där modeller använder **neural networks** för att efterlikna mänsklig hjärnaktivitet. Ett **Artificial Neural Network (ANN)** består av flera lager av noder (**neurons**), organiserade i:

* **Input Layer**
* **Hidden Layers**
* **Output Layer**

Varje neuron tar emot en viktad summa av input-data och passerar den genom en **activation function** som bestämmer om informationen ska skickas vidare. Vanliga aktiveringsfunktioner är:

* **ReLU (Rectified Linear Unit)**
* **Sigmoid**
* **Softmax** (för multiclass classification)

Djupa neurala nätverk används ofta inom **bildigenkänning, NLP (Natural Language Processing)** och **talförståelse**. De kan tränas med **backpropagation** och **gradient descent**.

**Azure Machine Learning**

Azure Machine Learning är en molnbaserad plattform från Microsoft som erbjuder verktyg för att bygga, träna och hantera ML-modeller. Plattformen har funktioner som:

* **AutoML**, vilket automatiserar valet av bästa modell.
* **Azure Machine Learning Studio**, en användarvänlig gränssnittsmiljö.
* **Experiment Tracking**, där träningssessioner kan loggas och jämföras.
* **Model Deployment**, där modeller enkelt kan distribueras som webbtjänster.

Azure Machine Learning kan integreras med populära ML-ramverk som **TensorFlow**, **PyTorch** och **scikit-learn**. Genom att använda molnet kan man utnyttja skalbara beräkningsresurser och säkerställa replikering av experiment.

**Slutsats**

Machine Learning har blivit en kritisk del av modern teknologi, med tillämpningar inom allt från prediktiv analys till autonoma system. För att skapa effektiva modeller är det viktigt att förstå olika ML-paradigmer, välja rätt algoritm och använda verktyg som Azure Machine Learning för att hantera modellens livscykel. Med rätt metodik och resurser kan ML användas för att effektivisera beslutsfattande, automatisera processer och skapa mer intelligenta system.