

# **Edge AI Portfolio**

Yves Braeckman

2 januari 2026

# Inhoudsopgave

<b>1 Inleiding</b>	<b>3</b>
<b>2 Portfolio: Lessen en Labo's</b>	<b>4</b>
2.1 Leren uit data . . . . .	4
2.2 Data . . . . .	4
2.3 Data exploratie . . . . .	5
2.4 Manipuleren van data . . . . .	5
2.5 Trainen en testen . . . . .	6
2.6 Labo 1 . . . . .	6
2.7 Labo 2 . . . . .	7
2.8 Labo 3 . . . . .	7
2.9 Labo 4 . . . . .	7
2.9.1 Beschrijving en Aanpak . . . . .	7
2.9.2 Persoonlijke Inzichten en Reflectie . . . . .	7
<b>3 Edge AI Project: Rapportage</b>	<b>8</b>
3.1 Projectintroductie . . . . .	8
3.2 Eigen Bijdrage en Leerproces . . . . .	9
3.2.1 Wat heb ik bijgedragen? . . . . .	9
3.2.2 Wat heb ik geleerd? . . . . .	9
<b>A Appendix: Technical Project Guide</b>	<b>10</b>
A.1 Benodigde Hardware . . . . .	10
A.2 Software en Dependencies . . . . .	10
A.3 Installatie-instructies . . . . .	10
A.4 Opstartprocedure . . . . .	10
A.5 Testinstructies en Model Performantie . . . . .	11
A.5.1 Model Analyse . . . . .	11
A.5.2 Foutafhandeling en Beperkingen . . . . .	11

# 1 Inleiding

Dit document vormt het portfolio voor het opleidingsonderdeel Edge AI.

De wereld van AI evolueert razendsnel. Waar de focus vroeger vaak lag op zware cloudberekeningen, zien we nu een verschuiving naar lokale intelligentie. Om deze technologie effectief toe te passen, is echter meer nodig dan enkel het kunnen schrijven van code. Een diepgaand inzicht in de volledige data-pipeline, van het verzamelen en opschonen van ruwe data tot het trainen, evalueren en deployen van modellen, is essentieel.

Dit portfolio is opgebouwd uit twee grote delen die deze vaardigheden weerspiegelen:

1. **Deel 1: Lessen en Labo's:** In dit deel reflecteer ik op de theoretische hoofdstukken en de uitgevoerde labo-opdrachten. Hierbij ligt de nadruk niet op het louter opsommen van definities, maar op mijn persoonlijke inzichten. Ik beschrijf hoe mijn begrip van concepten als data-exploratie, manipulatie en model-training is veranderd en welke valkuilen (zoals *garbage in, garbage out*) ik heb leren herkennen.
2. **Deel 2: Het Project:** Het tweede deel focust op de praktische integratie van deze kennis in het groepsproject. Hierin hebben we een afvalsorteersysteem ontwikkeld met behulp van Computer Vision op een Google Coral Dev Board. Dit deel demonstreert de vertaalslag van theorie naar een werkend prototype.

Met dit verslag wil ik aantonen dat ik niet alleen de technische vaardigheden beheers om AI-modellen te bouwen, maar ook de kritische mindset heb ontwikkeld die nodig is om data correct te interpreteren en verantwoorde modellen te ontwikkelen.

## 2 Portfolio: Lessen en Labo's

### 2.1 Leren uit data

Bij het doornemen van dit hoofdstuk werd me eigenlijk pas echt duidelijk hoe groot het verschil is tussen hoe wij als mensen leren en hoe we dat proberen na te bootsen in machines. Ik vond het heel interessant om te zien hoe een ML model eigenlijk een wiskundige definitie is van een functie die de data zo goed mogelijk probeert te benaderen.

Ik dacht vroeger dat een AI gewoon weet wat iets is, maar het hoofdstuk over de MNIST-dataset liet goed zien dat het eigenlijk draait om waarschijnlijkheden. Het model zegt niet dit is een 6, maar ik ben 99,9 zeker dat dit een 6 is.

Tot slot is het onderscheid tussen ML en AI voor mij nu duidelijker. ML is het herkennen van het patroon, AI is de stap erna: er daadwerkelijk een autonome actie aan koppelen.

Ik heb geleerd dat machine learning eigenlijk gewoon wiskunde is die probeert de werkelijkheid te imiteren, zonder dat het model echt 'begrijpt' wat het ziet. Het belangrijkste inzicht dat ik meeneem is dat we altijd rekening moeten houden met onzekerheid omdat het altijd een benadering van de realiteit blijft. Nu ik ook het verschil tussen het model en de toepassing scherp heb, voel ik me beter voorbereid om met de praktijkopdrachten aan de slag te gaan.

### 2.2 Data

De term "Garbage In, Garbage Out" kende ik al wel, maar dit hoofdstuk heeft die term voor mij veel concreter gemaakt. Ik dacht eerst dat het vooral ging over 'foute' getallen, maar ik begrijp nu dat het probleem eerder is: begrijp ik eigenlijk wel waar ik naar kijk en begrijp ik waarover ik verwacht dat het ML model een voorspelling maakt?

Een echte eye-opener was voor mij de sectie over het vinden van externe data. Ik had er niet bij stil gestaan dat een goede dataset vinden echt een uitdaging op zich is. Het idee dat je kritisch moet zijn over de bron, de meetmethode en zelfs de context van de data is iets wat ik zeker ga meenemen in mijn project verder in het semester.

De uitleg over datatypes vond ik best complex. Voorheen zag ik data gewoon als text of getal. Het onderscheid tussen ordinaal, nominaal en intervallschalen was nieuw voor mij. Vooral het stukje over circulaire data vond ik interessant: je kunt inderdaad niet zomaar het gemiddelde nemen van 23:00 uur en 01:00 uur, want dan kom je wiskundig op de middag uit, terwijl het middernacht moet zijn.

Ik heb ook het belang van het codeboek ingezien. Zonder dat document kan je de cijfers in een dataset vaak niet correct interpreteren.

Mijn belangrijkste inzicht uit dit hoofdstuk is dat de kwaliteit van mijn AI-model al bepaald wordt ver voordat ik de eerste regel code schrijf. Het verzamelen, begrijpen en opschonen van data is misschien wel de belangrijkste stap. Ik heb geleerd dat ik kritisch moet kijken naar databronnen, rekening moet houden met de specifieke eigenschappen van variabelen en altijd op zoek moet gaan naar het codeboek om de context van de cijfers te begrijpen. Zonder die basis is het niet verantwoord om verder te gaan.

## 2.3 Data exploratie

Data exploratie leek me vooraf vooral snel wat grafiekjes maken om te zien hoe de data eruit ziet. Dit hoofdstuk liet me zien dat het eigenlijk de belangrijkste fase is om fouten in een later stadium te voorkomen. De DIPKEVER-checklist vond ik een handig ezelsbruggetje. Het voorbeeld van de terrorisme-dataset liet zien hoe makkelijk je misleid kan worden: als een jaartal als een float wordt ingelezen in plaats van een int, kan dat later voor rare bugs zorgen.

Het concept dat de meeste indruk maakte, was de log-transformatie bij de behandelingsduur. In de eerste grafiek zag de data er enorm scheef uit, maar door de natuurlijke logaritme te nemen, veranderde dit in een normaalverdeling. Dit liet me zien dat data soms 'verstopt' zit en dat je wiskundige technieken nodig hebt om het bruikbaar te maken voor statistische modellen.

Exploratory Data Analysis is voor mij verschoven van grafieken maken naar detectivewerk. Ik heb geleerd dat ik eerst de structuur (datatype, missende waarden) moet controleren voordat ik inhoudelijk ga analyseren. Mijn belangrijkste inzicht is dat visualisatie niet alleen dient om data te presenteren, maar vooral om dataproblemen zoals scheve verdeelingen of outliers op te sporen die mijn model kunnen verpesten. Ik ga voortaan zeker de DIPKEVER-regels toepassen en kritischer kijken naar hoe data verdeeld is voordat ik verder ga.

## 2.4 Manipuleren van data

Wat ik vooral interessant vond in dit hoofdstuk, is de nuance bij het slicen van de dataset. Het selecteren van variabelen (kolommen) doen we continu, maar het filteren van instanies (rijen) is iets waar we voorzichtig mee moeten zijn. In de theorie werd benadrukt dat je rijen niet zomaar mag weggooien, tenzij er een gegronde reden is, zoals duidelijke invoerfouten of privacykwesties.

Ik heb geleerd dat transformaties soms noodzakelijk zijn om systemen met elkaar te laten praten. Simpele dingen zoals het aanpassen van encodings, het trimmen van spaties in tekstvariabelen of het toevoegen van voorloopnullen lijken simpel, maar zijn cruciaal om data te standaardiseren. Zonder deze stap kunnen ogenschijnlijk gelijke ID's door de computer als verschillend worden gezien, wat analyses fout laat lopen.

Tot slot hebben we gekeken naar het herschalen van numerieke waarden. Vooral normalisatie en standaardisatie zijn technieken die ik zeker ga toepassen in toekomstige opdrachten, omdat veel Edge AI-algoritmes slechter presteren als de schaal van variabelen te ver uit elkaar ligt.

Conclusie In dit onderdeel heb ik geleerd hoe ik ruwe data kan omzetten naar een bruikbare dataset voor analyse. Ik heb in de praktijk gebracht hoe ik uitschieters kan identificeren en verwijderen met behulp van visualisaties zoals de density plot en booleaanse maskers. Mijn belangrijkste resultaat is het inzicht dat elke manipulatie, of het nu gaat om het weglaten van een rij of het afronden van een getal, informatieverlies kan betekenen. Het is dus mijn taak om deze keuzes transparant te rapporteren en te verantwoorden, zodat de analyse integer blijft.

## 2.5 Trainen en testen

Het belangrijkste inzicht uit dit hoofdstuk was voor mij de realisatie dat Machine Learning geen exacte wetenschap of toverkunst is waarbij we altijd het gewenste resultaat bekomen. We moeten leren leven met onzekerheid en onze verwachtingen bijstellen van "we zullen voorspellen" naar "we hopen te kunnen voorspellen".

Ik heb geleerd dat strengheid hierbij cruciaal is. De test-set moet volledig verborgen blijven voor het algoritme en eigenlijk ook voor mij tot het allerlaatste moment. Het was leerzaam om te zien dat er naast de training- en test-set vaak nog een validatie-set nodig is.

Een interessant detail vond ik de uitzondering voor tijdreeksen. Waar we normaal data willekeurig door elkaar gooien om een goede split te krijgen, mag dit absoluut niet bij data met een tijdscomponent vanwege autocorrelatie. Hier moeten we echt een knip in de tijd maken, bijvoorbeeld alles na een bepaald jaartal gebruiken als test.

De workflow voor deep learning gaf mij veel inzicht in het concept overfitting. Door te kijken naar de loss van zowel de training- als de validatie-set tijdens de epochs, kun je precies zien wanneer het model stopt met leren en begint met 'uit het hoofd leren'.

Mijn belangrijkste conclusie is dat een model pas waarde heeft als het zich heeft bewezen op data die het nog nooit heeft gezien en dat ik als ontwikkelaar altijd waakzaam moet zijn voor valkuilen zoals data lekkage die mijn resultaten kunnen vertekenen

## 2.6 Labo 1

Ik heb in dit labo geleerd hoe ik een dataset kan inladen en verkennen met behulp van Python en Pandas. Het was interessant om te zien hoe ik met slechts een paar regels code al een goed beeld kon krijgen van de structuur en inhoud van de data.

Ik heb geleerd hoe ik missende waarden kan identificeren en hoe ik basisstatistieken kan berekenen om inzicht te krijgen in de distributie van de variabelen.

Daarnaast heb ik geleerd hoe ik visualisaties kan maken met Matplotlib om patronen en relaties in de data te ontdekken. Door een correlatiematrix en heatmap te maken, kreeg ik inzicht in welke variabelen daadwerkelijk invloed hadden op de te voorspellen variabele. Dit bevestigde voor mij het nut van stap 2 uit de opdracht: zonder deze visuele check tast je eigenlijk in het duister over welke features relevant zijn.

Mijn belangrijkste inzicht is dat een grondige verkenningsanalyse essentieel is voordat ik verder ga met analyse of modellering. Het helpt me om potentiële problemen te identificeren en beter geïnformeerde beslissingen te nemen over hoe ik de data verder wil verwerken. Dit is ook nog eens extra duidelijk geworden tijdens het Hack The Future event, waar ik een ML opdracht had. Hier heb ik de vaardigheden die ik in dit labo heb geleerd kunnen toepassen om de dataset te verkennen en voor te bereiden voor modellering.

## 2.7 Labo 2

## 2.8 Labo 3

Ik heb in dit labo geleerd hoe een belangrijke rol ML speelt in het dagelijkse leven. Het was heel interessant om te zien hoe Google hun verzamelde data en ML modellen gebruikt om ons gepersonaliseerde advertenties te tonen en aankopen te voorspellen. Ik vond het in het begin lastig om te werken met de Google Big Query omgeving, maar na een tijdje had ik de basis onder de knie.

Ik heb geleerd hoe ik SQL queries kan schrijven om data op te halen en te analyseren. Het was interessant om te zien hoe ik met behulp van SQL verschillende inzichten kon verkrijgen uit de grote datasets die Google ter beschikking stelt.

## 2.9 Labo 4

### 2.9.1 Beschrijving en Aanpak

**Aanpak:** Tijdens het maken van het labo heb ik volgende stappen doorlopen:

1. **Data Acquisitie en Preprocessing:** Als eerste stap heb ik scaling toegepast op de dataset zodat alle afbeeldingen van uint8 [0, 255] naar float32 [0.0, 1.0] werden omgezet.
2. **Training en testen:** De tweede stap was het verdelen van de data in een training stuk en een test stuk om op het einde de performantie te kunnen testen.
3. **Model bouwen:** Na een beetje opzoekwerk had ik gelezen dat een Conv2D laag beter werkt voor afbeeldingen dan een Dense laag. Ik heb vervolgens verschillende combinaties van lagen geprobeerd en verschillende groottes van lagen.
4. **Training en Compilatie:** Het model is gecompileerd met de Adam optimizer. Ik heb hier gekozen voor een heel lage learning rate omdat het model anders stopte met leren. Het model is getraind gedurende 15 epochs.

### 2.9.2 Persoonlijke Inzichten en Reflectie

Ik vond dit labo interessant omdat dit alles van de vorige labo's een beetje samenbracht. Ik vond het wel lastig om het model te bouwen omdat we hier niet veel over gezien hebben. Dit was dus een trial-en-error process en liep niet vlot.

Ik heb uiteindelijk online opgezocht en gebruik gemaakt van LLM's om te begrijpen wat alle parameters doen bij het bouwen van een model. Zo heb ik bijgeleerd over de learning rate en het belang van dit getal bij de training. Wanneer ik dit te hoog koos, ging het model in volledige chaos en leerde het niet, maar als ik het laag koos bleef het model steken op een bepaalde accuracy.

## 3 Edge AI Project: Rapportage

### 3.1 Projectintroduction

#### Motivatie

Voor het project van Edge AI, heb ik, samen met mijn groep, gekozen om een afvalsorteersysteem te maken op basis van machine learning. Ik vond dit onderwerp interessant omdat dit een brug bouwt tussen sustainable IT en Edge AI, we gebruiken IT om duurzamer te zijn. Ook vond ik het mooi een dagelijks probleem te kunnen aanpakken, aangezien er vaak fouten tegen gemaakt worden en het veel efficiënter zou kunnen wanneer dit automatisch gebeurt.

#### Doel

Het doel van dit project is het ontwerpen en trainen van een machine learning model dat in staat is om vijf verschillende afvalcategorieën correct te classificeren. Daarnaast is het doel om dit model te implementeren op een Coral Dev Board. Hiermee willen we aantonen dat het systeem in staat is om op basis van visuele input een fysieke actie aan te sturen. Binnen dit project simuleren we de aansturing van een sorteermachine visueel met behulp van vijf LED's, waarbij elke LED correspondeert met een specifieke afvalstroom.

#### Aanpak

Voor de praktische uitwerking van het project is er gebruik gemaakt van Roboflow voor het uitbreiden en cleanen van de dataset, Python met het TensorFlow library voor het ontwerpen en trainen van het model en een Coral Dev Board met een webcamera.

#### Opzet van het systeem

De technische structuur van het systeem is opgebouwd als een eindeloze lus in Python die op het Coral Dev Board draait.

1. **Initialisatie:** Bij het opstarten laadt het script het geconverteerde .tflite-model in het geheugen van de Edge TPU en worden de GPIO-pinnen voor de vijf LED's geconfigureerd als output.
2. **Input** De lus begint met het ophalen van een frame van de live webcam-feed. Dit beeld wordt voorbewerkt door het te schalen naar de input-resolutie die het model verwacht.
3. **Inferentie** De Edge TPU voert de berekeningen uit op het beeld. Het resultaat is een lijst met waarschijnlijkheidsscores voor elk van de vijf afvalcategorieën.
4. **Output:** De code controleert of de hoogste score boven een vooraf ingestelde drempelwaarde ligt. Als een van de score's boven de drempel ligt, dan wordt via de GPIO-interface de LED aangestuurd die overeenkomt met de herkende categorie. Indien niets met zekerheid wordt herkend, blijven de LED's uit.

Dit proces herhaalt zich heel snel, wat zorgt voor een real-time reactie wanneer een gebruiker afval voor de camera houdt.

## **3.2 Eigen Bijdrage en Leerproces**

**3.2.1 Wat heb ik bijgedragen?**

**3.2.2 Wat heb ik geleerd?**

# A Appendix: Technical Project Guide

## A.1 Benodigde Hardware

Hardware Component	Specificatie / Opmerking
Google Coral Dev Board	Om code op te draaien
USB-C Kabel	Voor stroomtoevoer
Ethernet Kabel	Voor SSH
USB Webcam	Input voor beeldherkenning
5x LED	Rood, Groen, Geel, Blauw, Wit
5x Weerstand	330Ω
Breadboard	Voor het prototype circuit
Jumper Wires	Male-to-Male / Male-to-Female
PC / Laptop	Voor SSH-toegang en opstarten

Tabel 1: Lijst van benodigde hardware

## A.2 Software en Dependencies

## A.3 Installatie-instructies

## A.4 Opstartprocedure

### 1. Hardware Opstelling

Voordat het bord wordt ingeschakeld, moeten alle componenten correct verbonden zijn:

1. **Camera:** Sluit de USB-webcam aan op de USB-A poort van het Coral Dev Board. Richt de camera op een witte achtergrond met voldoende belichting.
2. **Circuit:** Sluit de LED's en weerstanden aan op het breadboard en verbind deze met de GPIO-pinnen volgens het bedradingsschema.
3. **Netwerk:** Verbind de ethernetkabel met het bord en laptop voor SSH.
4. **Stroom:** Sluit als laatste de USB-C voedingskabel aan op de poort met het label PWR.

### 2. Verbinding en Uitvoering

#### Stap A: Verbind via SSH

Open een terminal en maak verbinding met het bord:

```
ssh mendel@192.168.1.100
```

#### Stap B: Navigeer naar de projectmap

Ga naar de folder waar de broncode en het model staan:

```
cd /home/mendel/Edge-AI-Project
```

### **Stap C: Start de applicatie**

Voer het Python-script uit om de detectie te starten.

```
python3 main.py
```

### **Stap D: De applicatie gebruiken**

Ga op de laptop/PC naar de browser en surf naar 192.168.1.100:5000

## **3. Stoppen van het systeem**

Om het programma te stoppen, druk je in de terminal op:

CTRL + C

## **A.5 Testinstructies en Model Performantie**

### **A.5.1 Model Analyse**

### **A.5.2 Foutafhandeling en Beperkingen**