

converting_format

October 16, 2025

1 Markdown/Notebook naar PDF: integratie van terminal en Pandoc in JupyterLab

Dit bewijsstuk heeft betrekking op criterium 2 (nieuwverworven kennis en vaardigheden toepassen) en criterium 11 (verantwoordelijkheid voor de eigen ontwikkeling).

In dit bewijsstuk laat ik zien dat ik oefen met mijn nieuwe werkomgeving (JupyterLab) door mijn bewijsstukken niet standaard in Word, maar in een Jupyter Notebook te schrijven. Aangezien .ipynb-bestanden niet rechtstreeks kunnen worden ingeleverd in Portflow, moest ik daarbij ook leren hoe ik deze kan omzetten naar bijvoorbeeld .pdf.

In Rstudio kon ik eenvoudig exporteren naar PDF, maar binnen JupyterLab bleek dit niet zo te zijn. Ik ben daarom gaan uitzoeken hoe dit conversieproces werkt, welke tools daarvoor nodig zijn, en hoe ik de tegengekomen problemen heb opgelost.

1.0.1 Installatie en conversie

Ik heb twee verschillende methoden gevonden om Jupyter Notebooks naar PDF te converteren:

1. Pandoc - een commandline tool waarmee je verschillende bestandsformaten kunt omzetten.
2. Jupyter nbconvert - een ingebouwde Jupyter-functie die gebruikmaakt van LaTeX om PDF-bestanden te genereren

Na testen bleek `nbconvert` beter geschikt omdat dit ook codecellen en de bijbehorende uitvoer netjes mee neemt in het PDF-bestand. Via `pandoc` zie je uiteindelijk alleen de markdowncellen terug.

1.0.2 Controle installatie en eerste test met Pandoc

```
[ ]: # controleer huidige directory (zorg er voor dat je in de map zit waar het
      ↳bestand te vinden is dat je wilt converteren)
pwd

# controleer of Pandoc is geïnstalleerd
pandoc --version

# converteer notebook naar PDF
pandoc -o converting_format.pdf converting_format.ipynb
```

Het PDF-bestand werd succesvol aangemaakt, maar ik merkte dat de codecellen niet zichtbaar waren in het eindresultaat. Ook zag de opmaak er niet optimaal uit. Pandoc blijkt vooral bedoeld te zijn voor Markdown (.md) en niet voor de notebook structuur.

1.0.3 PDF export via Jupyter nbconvert

Tijdens het omzetten van het notebook naar PDF met `nbconvert` kwamen verschillende LaTeX-fouten voor. Eerst probeerde ik losse LaTeX-pakketten te installeren met behulp van chatGPT, maar er ontstonden fouten zoals:

- `! LaTeX Error: File 'eurosym.sty' not found.`
- nog andere ontbrekende pakketten die nodig zijn voor notatie van tekens (bijvoorbeeld “€”) en opmaak

Uiteindelijk kreeg ik het advies om een volledige LaTeX-distributie te installeren (TinyTeX volledig), waardoor alle benodigde pakketten automatisch beschikbaar kwamen. Het is nu gelukt om een notebook om te zetten naar een pdf-bestand met het volgende commando.

```
[ ]: ### Converteren met nbconvert  
jupyter nbconvert --to pdf converting_format.ipynb
```

Het viel me direct op dat ook de opmaak veel beter is. De codecellen worden netjes weergegeven, er zijn geen grote lege ruimtes aan de zijkanten meer, en de inhoud wordt goed verdeeld over meerdere pagin's, wat het geheel overzichtelijker maakt.

De LaTeX-library die gebruikt wordt door `nbconvert` biedt ook veel mogelijkheden om de layout van het bestaande notebook aan te passen. Dit was eigenlijk het volgende punt waar ik naar wilde kijken. Denk bijvoorbeeld aan lettertype, grootte en kleur van tekst en code, maar ook aan stijlen voor tabellen, figuren en speciale elementen zoals formules.