

Tekoälyllä automaatiota korkeakoulukirjastojen lainausdatan käsittelyyn

29.12.2025

Aineiston lainaaminen on kirjastojen ydinpalvelua. Lainaamisesta kertyvällä datalla on keskeinen merkitys muun muassa aineiston käytön seurannassa ja kokoelmien kehittämisessä. Kirjastojärjestelmien tallentamaa dataa käytetään muuhunkin kuin kirjaston omiin tarkoituksiin. Tällainen on esimerkiksi lainauskorvauksen maksaminen.

Mikä lainauskorvaus?

Lainauskorvaus on tekijänoikeuskorvaus, joka maksetaan teosten ilmaisesta käyttämisestä ja siitä tekijälle aiheutuvasta tulonmenetyksestä. Yleiset ja korkeakoulukirjastot toimittavat vuosittain Sanastolle tiedot lainatuista teoksista ja niiden lainamääristä. Näiden tietojen perusteella opetus- ja kulttuuriministeriön hyväksymänä Sanasto tilittää valtion budjetista maksettavat korvaukset, jotka vuositasolla ovat olleet yli 10 miljoonaa euroa (Sanasto, n.d.).

Millainen laina oikeuttaa lainauskorvaukseen?

Korkeakoulukirjastosta ulos lainattu kirja, nuotti, äänite tai äänikirja huomioidaan lainauskorvausta laskettaessa. Laina-ajan pituudella ei ole merkitystä. Myös uusinnat lasketaan mukaan. Lainojen uusiminen eli laina-ajan pidentäminen voidaan toteuttaa joko kirjaston henkilökunnan toimesta tai asiakkaan itsenäisesti

suorittamana. Lisäksi uusintoja on mahdollista tehdä automaattisesti määriteltyjen ehtojen mukaisesti.

Kuinka lainausdataa kerätään?

Lainausdatan kerääminen Sanastolle voisi ajatella olevan aika selkeää ja suoraviivaista nykyaikaisissa kirjastojärjestelmissä. Tämä voidaan tehdä esimerkiksi SQL-kyselyllä, valmiilla tilastoskriptillä tai kirjastojärjestelmän raportointityökalulla. Ei sen kummempaa noin periaatteessa. Joskus järjestelmän muutokset vaikeuttavat asiaa ja kriittinen data tuntuun katoavan tai muuttuvan vaikeammin hyödynnettävään muotoon (Múnera Willes, 2019).

Mahdolliset taloudelliset vaikutukset

Automaattisiin uusintoihin liittyvät aikaleimamuutokset herättivät ymmärrettävästi huolta kirjastoammattilaisten keskuudessa. SEAMK Kirjaston osalta maksettavia lainauskorvauksia voisi jäädä vuodessa noin 4000 € tilittämättä, jos automaattiuusintoja ei voitaisi laskea mukaan lainausmääriin. Kerrannaisvaikutus maksamattomiin korvauksiin muodostuisi huomattavasti suuremmaksi, jos kaikki samaa kirjastojärjestelmää käyttävät kirjastot eivät automaattiuusintoja kykenisi ilmoittamaan (Helsingin yliopisto XWiki, n.d.).

Onko tarvittavaa dataa?

Kirjastojärjestelmämme datassa oli viitteitä siitä, että automaattiuusinnoista jäisi kuitenkin jokin jälki tietokantaan. Olisiko tähän joku toteuttanut jonkin ratkaisun maailmalla, kuten joskus on saattanut olla? Valmista ratkaisua ei vain kuitenkaan tuntunut löytyvän.

Järjestelmän tietomallin takia lainaus- ja uusintadata ei ollut enää yhdistettävissä järjestelmän omilla työkaluilla. Sinänsä tämä ei ole mitenkään erikoista, että dataa tuodaan ulos tietokannoista toisiin järjestelmiin. Kirjastojen tietojärjestelmät tarjoavat myös erilaisia avoimia rajapintoja, joiden kautta voidaan toteuttaa sellaisia toimintoja ja palveluita mitä järjestelmät eivät itse tarjoa.

Lähes mahdottomasta mahdollista

Seuraavaksi haasteena oli datan yhdistäminen mahdollisimman suoraviivaisesti ja yksinkertaisesti. Aikaisempien kokemusteni perusteella Python-ohjelmointikieli voisi tarjota sopivan ratkaisun tiedostojen ja tietojen yhdistämiseen.

Pythonin alkeisiin olen tutustunut, mutta toimivaa koodia tuskin osaisin tyhjästä tehdä. Myöskään osaavan koodarin käyttö ei ollut mahdollista. Onneksi erilaisia koodaamista mahdollistavia ja helpottavia työkaluja on olemassa, kuten GitHub Copilot ja ChatGPT.

Lyhyen pohdinnan jälkeen valitsin kaverikseni GitHub Copilotin, joka yhdistettynä Visual Studio Coden kanssa vaikutti kaltaiselleni koodausta harjoittelevalle sopivalta yhdistelmältä. Visual Studio oli minulle jonkin verran

tuttu aiempien projektien kautta.

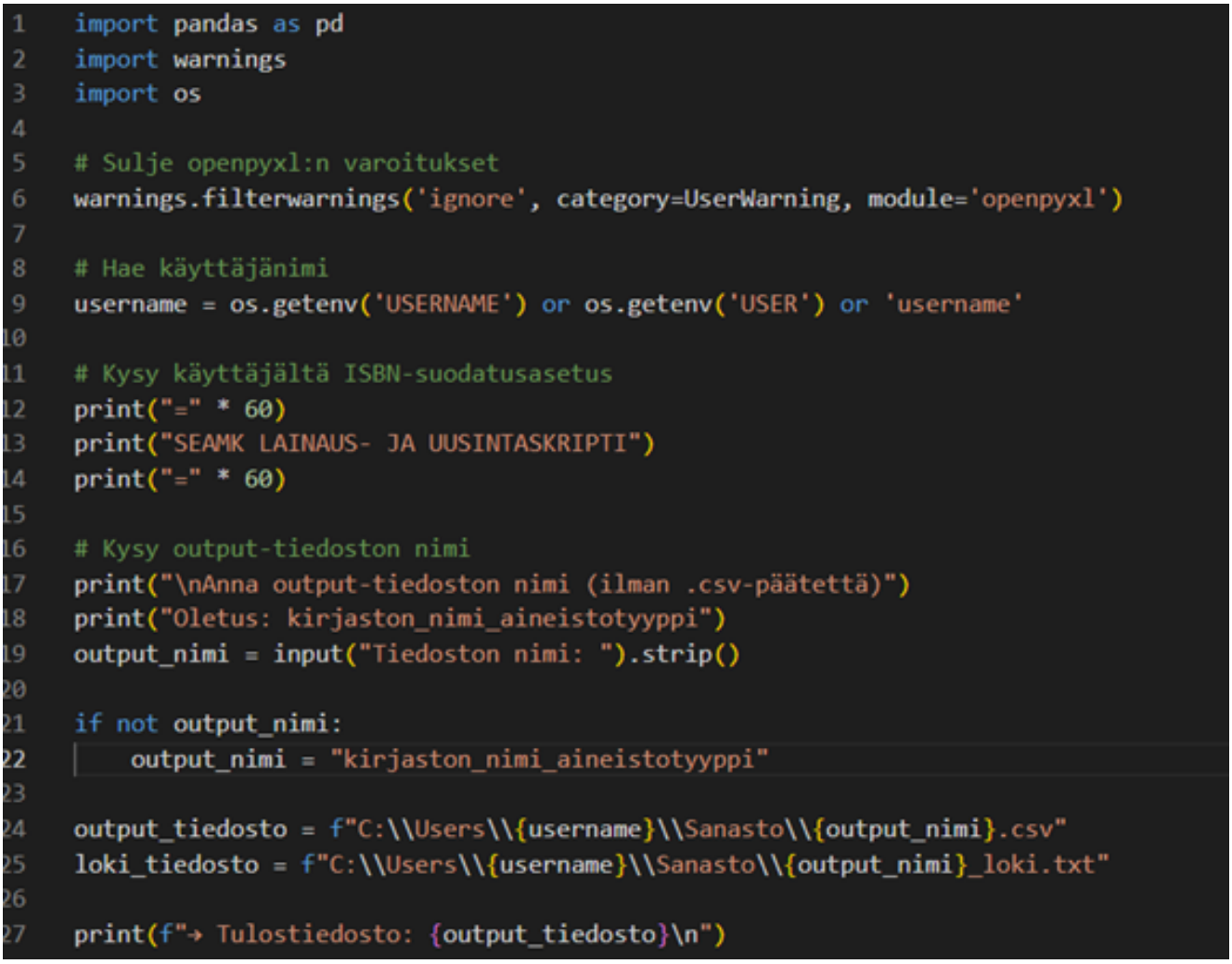
Työskentelyä GitHub Copilotin kanssa

Tuloksen saavuttamiseksi koodia kehitettiin vaiheittain iteratiivisesti. Tavoitteena oli automatisoida Excel-muodossa olevien tiedostojen yhdistämisprosessi Python-skriptillä. Kun yksi vaihe saatiin sellaiseksi, että sitä voitiin hyödyntää seuraavassa vaiheessa, päästiin eteenpäin. Kirjoitin kehotteen toisensa perään ja Copilot huolehti koodaamisesta aina siinä välillä väsymättä. Koodausprojekti eteni yllättävän sujuvasti ja peruskoodi valmistuikin muutamassa tunnissa. Ei ongelmitta, mutta kuitenkin.

Vähitellen tekninen toiminnallisuus jalostui interaktiiviseksi, intuitiiviseksi ja dokumentoiduksi kokonaisuudeksi. Tässä kaikessa GitHub Copilot oli korvaamaton apuri. Lähdedataan liittyviä puutteita tai ongelmia se ei voinut tietenkään täysin tunnistaa, mutta autoin sitä siinä. Koodia se osasi tehdä yhteisymmärryksessä kehotteideni kanssa ja kirjoittaa käyttöohjetta minua nopeammin. Antoipa se vielä ideoita jatkokehittämiseenkin muun muassa koodin ajastamiseen, graafisen käyttöliittymän tekemiseen ja Mac/Linux-yhteensopivuuden toteuttamiseen.

Kun pyysin Copilotia vielä lopuksi tarkistamaan tehtyä koodia Python-ohjelmoinnin periaatteiden ja hyvien käytänteiden mukaiseksi, teki se jälleen työtä käskettyä. Lopputulosta en kykene arvioimaan kovinkaan syvällisesti, mutta siihen on vain luotettava. Ainakin skripti tekee sen, mihin se oli ajateltu.

Yhteiseksi hyödyksi



```python
import pandas as pd
import warnings
import os

# Sulje openpyxl:n varoitukset
warnings.filterwarnings('ignore', category=UserWarning, module='openpyxl')

# Hae käyttäjänimi
username = os.getenv('USERNAME') or os.getenv('USER') or 'username'

# Kysy käyttäjältä ISBN-suodatusasetus
print("=" * 60)
print("SEAMK LAINAUS- JA UUSINTASKRIPTI")
print("=" * 60)

# Kysy output-tiedoston nimi
print("\nAnna output-tiedoston nimi (ilman .csv-päätettä)")
print("Oletus: kirjaston_nimi_aineistotyyppi")
output_nimi = input("Tiedoston nimi: ").strip()

if not output_nimi:
    output_nimi = "kirjaston_nimi_aineistotyyppi"

output_tiedosto = f"C:\\Users\\{username}\\Sanasto\\{output_nimi}.csv"
loki_tiedosto = f"C:\\Users\\{username}\\Sanasto\\{output_nimi}_loki.txt"

print(f"→ Tulostiedosto: {output_tiedosto}\n")
```

Sanasto-skriptin Python-koodia (kuva: Oma kuvakaappaus, 2025).

Alkuperäisenä tavoitteena oli saada vain erillään oleva lainaus- ja uusintadata yhdistettyä hyödynnettävään muotoon Sanastoa varten. Toteutus piti kuitenkin yrittää tehdä niin, että samassa tilanteessa olevat kirjastot voisivat myös hyödyntää sitä. Sanasto-skripti on jaettu kirjastoille. Kehitystyötä voidaan nyt jatkaa yhteistyössä, aivan kuten korkeakoulukirjastot ovat tehneet tilastoinnin osalta jo pitkään.

Ratkaisu vaikuttaa käyttökelpoiselta ja toivon sen hyödyttävän automaattisia uusintoja käyttäviä kirjastoja ja erityisesti lainauskorvauksia saavia kirjailijoita ja kääntäjiä.

Jarkko Meronen
Suunnittelija, kirjastopalvelut
SEAMK

Kirjoittaja toimii suunnittelijana Seinäjoen ammattikorkeakoulun kirjastossa (SEAMK Kirjasto). Hänen vastuualueisiinsa kuuluvat muun muassa kirjastojärjestelmät, verkkopalvelut ja digitaalinen saavutettavuus.

Lähteet

Helsingin yliopisto XWiki. (n.d.). *Suomen Alma-kirjastot*. Haettu 16.12.2025, https://wiki.helsinki.fi/xwiki/bin/view/lumikko/Suomen%20Alma-kirjastot/

Múnera Willes, D. (2019, 14. elokuuta). *Get auto-renewals dates back into Analytics* [Idea submission]. Ex Libris Idea Exchange. https://ideas.exlibrisgroup.com/forums/308173-alma/suggestions/38372518-get-auto-renewals-dates-back-into-analytics

Sanasto. (n.d.). *Lainauskorvaus*. Haettu 16.12.2025, https://www.sanasto.fi/lainauskorvaus

Asiasanat: lainaustoiminta, tietojärjestelmät, automaatio, tekoäly, tietojenkäsittely, korkeakoulukirjastot, digitalisaatio, kirjastojärjestelmät