tkinter är ett inbyggt bibliotek i Python som används för att skapa grafiska användargränssnitt (GUI), som till exempel fönster och knappar. Det är ett bra val för utvecklare som vill skapa skrivbordsprogram eftersom det är lätt att använda och fungerar på alla större operativsystem utan att behöva installera extra programvara.

Koden gör det också möjligt att automatisera återkommande uppgifter och enkelt exportera data efter filtrering.

Excel är bra också för grundläggande datahantering, men erbjuder inte samma anpassningsmöjligheter eller programmeringsflexibilitet.

Här är en enkel förklaring av

vad koden gör:

Läser in en CSV-fil: Koden använder pandas-biblioteket för att läsa data från en CSV-fil.

Skapar ett användargränssnitt: Med hjälp av tkinter-biblioteket skapas ett interaktivt fönster som visar datan i en tabellform.

Filtrering: Användaren kan skriva in filter för att snabbt filtrera och se resultaten.

Exportfunktion: Det finns en knapp för att exportera den filtrerade datan till en ny CSV-fil.

Sammanfattning: Koden gör det möjligt att snabbt visa och analysera data med större flexibilitet och anpassningsmöjligheter än Excel, vilket är särskilt användbart när du behöver automatisera repetitiva uppgifter.

------------------------------------------------------------------------------------

**Kodexempel som fångade mitt intresse**

**här lite om kodem hur hanterar och visar data i en Tkinter-applikation**

**(Det här är bara korta rubriker för koden) :**

* Huvudram för data
* main\_frame = tk.Frame(root, bg="#EAEAEA")

Den här koden skapar en huvudram i applikationsfönstret med en ljusgrå bakgrundsfärg.

* **Visa data i tabellformat**
* self.tree = ttk.Treeview(main\_frame, columns=[col for col in df.columns], show='headings')
* self.tree.grid(row=1, column=0, padx=10, pady=10, sticky='nsew')

Här skapas en Treeview för att visa data i ett tabellformat. Kolumnerna sätts baserat på datafilens kolumnnamn, och tabellen visas i huvudrame

* Tillämpa filter
* def apply\_filter(self):

Denna funktion applicerar ett filter på datan baserat på användarens inmatning. Filtreringen gör att endast de rader som matchar det angivna kriteriet visas.

* Exportera data till CSV
* def export\_to\_csv(self):

Den här metoden exporterar den filtrerade datan till en CSV-fil. Användaren ombeds att välja en plats och ett filnamn, och datan sparas.

* Automatisk justering av kolumnbredd
* def auto\_adjust\_column\_width(self):

Denna funktion justerar automatiskt bredden på kolumnerna i tabellen baserat på det längsta innehållet i varje kolumn. Det säkerställer att all text visas korrekt utan att bli avklippt.