Teoretiska frågor

1. Lotta delar upp sin data i ”Träning”, ”Validering” och ”Test”, vad används respektive del för?

- **Tränings-data**
  Användning:
  Tränings-data används för att lära modellen att identifiera mönster och samband. Modellen justerar sina parametrar enligt de regler och metoder som bestäms av data-analytikern. Syftet är att minimera skillnader mellan modellens förutsägelser och de faktiska utfallen i tränings-data.

  Exempel:
  Om Lotta tränar en modell för exempelvis klassificera bilder på katter och hundar, kan tränings-data bestå av bilder där etiketten (katt eller hund) redan är känd, supervised learning.

- **Validerings-data**
  Användning:
  Validerings-data används för att utvärdera modellens prestanda under träningen för att justera inställningar som påverkar modellens beteende, exempelvis inlärningshastighet eller tröskelvärden för klassificering. Den används också för att övervaka modellens prestanda under träningen och för att undvika överanpassning (overfitting).

  Exempel:
  Lotta kan använda validerings-data för att avgöra vilken inlärningshastighet eller andra inställningar som ger bäst resultat för hennes modell.

- **Test-data**
  Användning:
  Test-data används för att utvärdera den färdigtränade modellens prestanda på data som modellen aldrig har sett tidigare. Detta ger en uppskattning av hur väl modellen generaliserar till ny och okänd data.

  Exempel:
  Efter att Lotta har tränat och finjusterat modellen, testar hon den på ett separat dataset med exempelvis bilder för att se hur väl den klassificerar katter och hundar i praktiken.

Sammanfattning av rollerna:

Dataset indelning	När används det?	Synlighet för modellen
Träningsdata	Under träning.	Modellen ser och använder träningsdata direkt.
Valideringsdata	Under utvärdering av modellen (till exempel för att anpassa hyperparametrar).	Modellen ser data indirekt, men den används inte för att justera parametrar.
Testdata	Endast efter träning är avslutad.	Modellen ser aldrig testdata förrän träningen är helt klar.

Att använda denna uppdelning är en grundläggande praxis för att säkerställa en robust och tillförlitlig modell.

2. Förklara (gärna med ett exempel): Ordinal encoding, one-hot encoding, dummy variable encoding.

- **Ordinal Encoding**
  Det är en metod för att representera kategoriska variabler som siffror. Kategorierna tilldelas unika heltal i en viss ordning, vilket innebär att siffrorna har en hierarkisk betydelse. Det används när kategorierna har en naturlig rangordning, exempelvis; Small, Medium, Large.

  Exempel:
  Om vi har en variabel som beskriver storlek:

  - Small --> 1
  - Medium --> 2
  - Large --> 3

- **One-Hot Encoding**
  En metod där varje kategori representeras som en binär (0 eller 1) kolumn. Varje rad har endast en kolumn med värdet 1, och resten är 0. Det används när kategorierna inte har en naturlig rangordning, exempelvis: Frukter, färger eller djur.

  Exempel:
  För att beskriva djur med kategoreierna "Cat", "Dog", "Rabbit" :

  - Cat → [1, 0, 0]
  - Dog → [0, 1, 0]
  - Rabbit → [0, 0, 1]
  - **Dummy Variable Encoding**
    Metoden liknar one-hot encoding, men en kategori (vanligtvis den första) utesluts. Detta minskar antalet kolumner med 1 och undviker problemet med multikollinearitet i vissa statistiska modeller. Det används ofta i regression och andra statistiska analyser där kategorier är oberoende.

    Exempel:
    För att beskriva djur med kategoreierna "Cat", "Dog", "Rabbit" :
    Här utesluts "Cat"

  - Cat → [0, 0]
  - Dog → [1, 0]
  - Rabbit → [0, 1]

3. Göran påstår att data antingen är ”ordinal” eller ”nominal”. Julia säger att detta måste tolkas. Hon ger ett exempel med att färger såsom {röd, grön, blå} generellt sett inte har någon inbördes ordning (nominal) men om du har en röd skjorta så är du vackrast på festen(ordinal) – vem har rätt?

Både **Göran och Julia** har rätt, men från lite olika perspektiv, vilket gör detta mer till en diskussionsfråga.

Göran påstår att datan är antingen ***ordinal*** eller ***nominal***. Detta är formellt korrekt men han förenklar klassificeringen av data till två distinkta typer.

- Nominal data: Kategoriska data utan inbördes rangordning (t.ex. färger, djurarter).
- Ordinal data: Kategoriska data med en naturlig ordning (t.ex. betyg som A, B, C).

Julia säger att data måste tolkas och detta är också korrekt eftersom kontexten kan avgöra hur en viss uppsättning av data ska tolkas. Hon lyfter fram att en variabels typ beror på hur den används och hur vi förhåller oss till den i analysen.

- Nominal: Färger (röd, grön, blå) har ingen naturlig rangordning om de bara beskriver en egenskap.
- Ordinal: Om kontexten ger en subjektiv ordning (t.ex. "röd skjorta gör dig vackrast"), så kan färgerna tolkas som ordinal data.

Julia och Göran påståenden är båda korrekta, men Julias poäng är djupare och visar vikten av att förstå datans kontext innan den klassificeras. Datatypen är inte alltid självskriven – hur data används i en specifik situation kan förändra dess tolkning.

4. Svara på frågan: Vad används joblib och pickle till?

Joblib och **Pickle** är båda bibliotek som används för att spara och ladda objekt i Python, vilket är särskilt användbart inom machine learning. Det gör det möjligt att lagra en modell efter träning och återanvända den för att göra förutsägelser utan att behöva träna om modellen från grunden.

Om man arbetar med mindre modeller eller Python-objekt generellt är **Pickle** att föredra. När man hanterar stora datamängder och komplexa ML-modeller som använder numpy-arrayer används främst **Joblib**.