1. **Träning**: Används för att lära modellen genom att justera dess parametrar baserat på träningsdata.

**Validering**: Används för att utvärdera modellens prestanda under träningen och hjälpa till att justera hyperparametrar, för att undvika överanpassning.

**Test**: Används för att bedöma modellens slutgiltiga prestanda på tidigare osedd data efter träningen.

1. **Ordinal Encoding**

**Definition**: En metod där kategoriska värden ersätts med heltal baserat på en ordning.

**Exempel**: Anta en kategori för utbildningsnivå:

* 1. ["Gymnasium", "Kandidat", "Master", "Doktor"]
  2. Ordinal Encoding: [1, 2, 3, 4]

Används när kategorier har en naturlig rangordning.

**One-Hot Encoding**

* **Definition**: Representerar varje kategori som en binär vektor där endast en position är "1" och resten är "0".
* **Exempel**: Anta en kategori för färger:
  + ["Röd", "Grön", "Blå"]
  + One-Hot Encoding:
    - Röd: [1, 0, 0]
    - Grön: [0, 1, 0]
    - Blå: [0, 0, 1]

Används när kategorier saknar inbördes rangordning.

**Dummy Variable Encoding**

* **Definition**: Liknar one-hot encoding, men en kategori utesluts för att undvika multikollinearitet vid linjära modeller.
* **Exempel**: Samma färger:
  + ["Röd", "Grön", "Blå"]
  + Dummy Encoding:
    - Röd: [1, 0]
    - Grön: [0, 1]
    - Blå (baslinje): [0, 0]

Baslinjekategorin (Blå här) representeras implicit och behövs inte som en egen

kolumn.

1. Julia har rätt.

Datans karaktär (ordinal eller nominal) beror på hur kategorierna tolkas i det specifika sammanhanget:

**Nominal data**: Kategorier utan inbördes rangordning, t.ex. {röd, grön, blå} när färger bara används för att beskriva egenskaper.

**Ordinal data**: Kategorier med en inbördes rangordning, t.ex. om {röd, grön, blå} representerar en subjektiv skönhetsgrad (t.ex. "röd är vackrast").

Julia påpekar att det är användningen av datan som avgör dess typ, vilket Göran inte tar hänsyn till.

1. **Joblib** och **pickle** används för att spara och ladda Python-objekt, inklusive tränade maskininlärningsmodeller, för att undvika att behöva träna om modellerna varje gång.

**Syfte**:

**Spara**: Efter att en modell tränats kan den sparas till en fil för framtida användning.

**Ladda**: Den sparade modellen kan laddas och användas direkt för att göra prediktioner utan att träna om modellen.

**Skillnader**:

**Pickle**:

Generisk metod för att serialisera (spara) och deserialisera (ladda) Python-objekt.

Kan hantera enkla och komplexa Python-objekt, men är långsammare för stora data.

**Joblib**:

Optimerad för att hantera stora datamängder och numpy-arrayer effektivt.

Rekommenderas för maskininlärningsmodeller som innehåller stora matriser.