Kunskapskontroll 2 Del 1

Besvara nedanstående teoretiska frågor koncist.

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

Träning

Används för att lära modellen de underliggande mönstren i datat.

Validering

Används för att finjustera modellens hyperparametrar och för att övervaka överträning (overfitting).

Test

Används för att utvärdera modellens slutliga prestanda.

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

Ordinal Encoding

Omvandlar kategorier till siffror baserat på en implicit ordning.

Exempel:

Kategori: {Låg, Medel, Hög}

Ordinal encoding: {Låg → 1, Medel → 2, Hög → 3}

Användning: När kategorierna har en meningsfull ordning (t.ex. betyg).

One-Hot Encoding

Skapar binära kolumner för varje kategori; 1 indikerar när en kategori är aktiv.

Exempel:

Kategori: {Hund, Katt, Fågel}

Hund: [1, 0, 0]

Katt: [0, 1, 0]

Fågel: [0, 0, 1]

Dummy Variable Encoding

Vad: Liknar one-hot encoding men tar bort en kolumn (referenskategori) för att undvika multikollinearitet i modeller som kräver detta, t.ex. regressionsmodeller.

Exempel:

Kategori: {Hund, Katt, Fågel}

Hund: [1, 0] Katt: [0, 1] Fågel (referens): [0, 0]

3. **Göran påstår att datan 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 en poäng, men Julia lyfter något viktigt: att tolkningen av data spelar en stor roll för att avgöra om den är ordinal eller nominal.

Göran har rätt om att data i sig är antingen nominal eller ordinal, men Julia poängterar korrekt att tolkningen och kontexten kan förändra hur vi ser på datan. I exemplet blir det alltså kontextberoende om färgerna är nominala eller ordinala.

4. **Läs följande länk: https://stackoverflow.com/questions/56107259/how-to-save-a-trained-model-by-scikit-learn (speciellt svaret från användaren som heter ”sentence”) som beskriver ”joblib” och ”pickle”. Det är alltså ett sätt att spara modeller och innebär att man kan träna en modell och sedan återanvända den för att göra prediktioner utan att behöva träna om modellen. Detta kommer ni ha nytta av om ni satsar på VG delen. Svara på frågan: Vad används joblib och pickle till?**

Pickle är ett bibliotek som serialiserar Python-objekt (gör om dem till en binär representation) för att kunna spara dem på disk eller skicka dem över nätverk. Man kan spara och ladda Python-objekt, som listor, dictionaryn, eller tränade modeller.

Joblib

Liknar Pickle men är optimerat för att serialisera stora objekt, särskilt NumPy-arrayer och modeller från scikit-learn. Joblib är snabbare än Pickle för stora objekt.

Joblib är effektivare vid hantering av objekt som innehåller stora numeriska data (t.ex. maskininlärningsmodeller. Det är vanligt förekommande inom maskininlärning att spara modeller och deras viktparametrar. Joblib är att föredra mellan dessa 2 alternativ. Även i joblib gör man en dump av sin modell.