Kunskapskontroll 2 – del 1

- Anita Jonsson -

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

Träningsdata avänds för att skapa och träna olika modeller

Valideringsdata används för att utvärdera modellerna

Testdatan används för att utvärdera modellens ”generaliseringsförmåga”. Att modellen inte ”overfit-ar”

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

Ordinal koding är när man ersätter t.ex färg med ett värde. Om färgen på jordgubben är grön blir får de värdet 1 i skalan och det är ett lägre värde än för färgen ljusröd.

Färg jordgubbe	Ordinalskala_färg
Grön	1
Ljusröd	2
Mörkröd	3

I one-hot encoding ersätter man kolumnen med en kolumn för varje kategori i variabeln. Har man variabeln kön och alternativen ,man – kvinna - vill ej uppge, skapas tre kolumner.

Variabel kön	Is_kvinna	Is_man	Is_vill ej uppge
Kvinna	1	0	0
Man	0	1	0
Vill ej uppge	0	0	1

För dummy variabel skapas en kolumn mindre än det finns kategorier. Den uteslutna kategorin framgår av de fall där varken Dummy_kvinna eller Dummy_man är sant.

Variabel kön	Dummy_kvinna	Dummy_man
Kvinna	1	0
Man	0	1
Vill ej uppge	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?

De har båda rätt. Samma typ av data kan vid ett tillfälle vara ordinal och vid ett annat nominal. Det måste man bestämma när man skapar sin modell hur informationen ska tillämpas. Färgen

på en frukt anger inte någon rangordning mellan olika frukter. Bananer är nödvädigtvis inte bättre än apelsiner för att deras färg är gul. Färg är då nominal. Medan färgen på samma sorts frukt kan ange en inbördes rangordning med hänsyn till hur mogen frukten kan förväntas vara. En gul banan är bättre än en grön banan. Färg är då ordinal.

4. Vad används joblib och pickle till?

Joblib och pickle kan båda användas till att spara en tränad/skapad model för att slippa träna om modellen varje gång man vill använda den. Detta bidrar till påskynda utvecklingsprocessen. Genom att spara ner olika modell kan man testa olika versioner mot varandra och det blir också ett sätt att versionshantera modeller.

Pickle har fördelen att den ingår i Pythons standard bibliotek men man blir samtidigt mer beroende av vilken Python version man använder. Pickle är inte lika bra som joblib på att hantera stora dataset och man rekommenderar därför ofta Joblib för machine leraning projekt.

Joblib är den modell som rekommenderas för machine learning projekt eftersom den jobbar snabbare på stora numpy arrays och den komprimerar (compress) filerna.

```
[14]: from sklearn import svm
      from sklearn import datasets

      ##########################
      # SAVE-LOAD using joblib #
      ##########################
      import joblib

      # save
      joblib.dump(extra_trees_clf, "model_mnist.joblib")
```

```
[14]: ['model_mnist.joblib']
```

Ref:

https://www.geeksforgeeks.org/understanding-python-pickling-example/

https://www.analyticsvidhya.com/blog/2023/02/how-to-save-and-load-machine-learning-models-in-python-using-joblib-library/