Teoretiska frågor

Besvara nedanstående teoretiska frågor koncist.

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

Träningsdata:
Detta är den största delen av data som används för att träna modellen. Modellen studerar datan, deras mönster och samband, och uppdaterar iterativt sina parametrar baserat på denna datamängd.

Valideringsdata:
Denna datamängd används under träningsprocessen för att utvärdera modellens prestanda på data som den inte har sett under träningen. Detta är nödvändigt för att: välja de optimala hyperparametrarna och finjustera modellen mer exakt, samt upptäcka modellens överanpassning och avsluta träningen.

Testdata:
Detta är data som hålls helt orörda tills modellen är helt tränad. De används för att ge en slutlig objektiv utvärdering av modellens effektivitet på helt nya data för modellen. Detta ger en uppfattning om hur bra modellen kommer att prestera i verkligheten.

En vanlig fördelning är ungefär 60 % träningsdata, 20 % valideringsdata och 20 % testdata.

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

Ordinal Encoding:
Används när kategorier har en naturlig ordning. Varje kategori ersätts med ett heltal.
Till exempel kan uppgifter på en Kanban-tavla i Trello ha följande prioriteringar:
Incident - 0
Hög prioritet - 1
Medel prioritet - 2
Låg prioritet - 3

One-Hot Encoding:
Skapar en ny binär kolumn för varje kategori. Varje rad får en 1 i sin kategorikolumn och 0 i övriga.
Jag använde den här metoden för att prediktera hus med ett värde på >= 500000 för datafilen housing.csv för en kategorikolumn.

Man kan säga att denna metod använder boolesk logik: True, False
*I detta exempel kunde man ha använt **Ordinal Encoding**, eftersom husens läge närmare strandlinjen har hög prioritet*

median_house_value	ocean_proximity_<1H OCEAN	ocean_proximity_INLAND	ocean_proximity_ISLAND	ocean_proximity_NEAR BAY	ocean_proximity_NEAR OCEAN	predicted_value
500001	0.0	0.0	0.0	1.0	0.0	179809.468442
500001	0.0	0.0	0.0	1.0	0.0	179815.573534
500001	0.0	0.0	0.0	1.0	0.0	180404.331600
500001	0.0	0.0	0.0	1.0	0.0	180440.199408
500001	0.0	0.0	0.0	1.0	0.0	180458.399204

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 14 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   longitude                  20640 non-null  float64
 1   latitude                   20640 non-null  float64
 2   housing_median_age         20640 non-null  int64
 3   total_rooms                20640 non-null  int64
 4   total_bedrooms             20433 non-null  float64
 5   population                 20640 non-null  int64
 6   households                 20640 non-null  int64
 7   median_income              20640 non-null  float64
 8   median_house_value         20640 non-null  int64
 9   ocean_proximity_<1H OCEAN  20640 non-null  float64
 10  ocean_proximity_INLAND     20640 non-null  float64
 11  ocean_proximity_ISLAND     20640 non-null  float64
 12  ocean_proximity_NEAR BAY   20640 non-null  float64
 13  ocean_proximity_NEAR OCEAN 20640 non-null  float64
dtypes: float64(9), int64(5)
memory usage: 2.2 MB
RMSE на тестовых данных: 116841.1289494461
Предсказания для домов стоимостью $500,000 и выше:
     predicted_value
0      179809.468442
...
990    180612.552854
991    180098.250952
```

Dummy Variable Encoding:

Liknar one-hot men droppar en av kategorierna (referenskategorin) för att undvika multikollinearitet. Skapar n-1 kolumner där n är antalet kategorier.

röd 0 0
grön 1 0
blå 0 1

När ska man använda vad?

- ✓ *Ordinal: För kategorier med naturlig rangordning*
- ✓ *One-hot: När kategorierna är oberoende och likvärdiga*
- ✓ *Dummy: För att undvika multikollinearitet, särskilt i linjär regression*

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 poänger, men Julias resonemang om att kontext spelar roll är särskilt viktigt att förstå. Data kan vara nominal eller ordinal beroende på hur den används och tolkas.
Julia påpekar dock att tolkningen av data kan variera beroende på sammanhang. I hennes exempel är färger normalt sett nominaldata, men i ett specifikt sammanhang (som att en röd skjorta gör någon "vackrast på festen") kan de tolkas som ordinaldata eftersom det finns en implicit rangordning.

Det är viktigt att inse att data inte alltid är strikt nominal eller ordinal utan kan vara kontextberoende.

Julia har rätt :)

4. Läs följande länk:
https://stackoverflow.com/questions/56107259/how-to-save-a-trainedmodel-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?

Joblib och Pickle används för att spara och ladda tränade maskininlärningsmodeller i Python.

Detta är särskilt användbart när:
- Träningen tar lång tid
- Du vill använda samma modell i olika applikationer
- Du behöver dela modellen med andra utvecklare

Den huvudsakliga skillnaden är:
Pickle:
- Ett inbyggt Python-bibliotek för att spara Python-objekt
- Kan användas för att spara alla typer av Python-objekt
- Passar bra för mindre modeller
Joblib:
- Specialiserat för att hantera stora numpy-arrayer
- Mer effektiv för stora dataset och modeller
- Snabbare än pickle för större filer

Huvudsyftet är att kunna:
- Spara en tränad modell till disk
- Återanvända modellen senare utan att behöva träna om den
- Dela modellen med andra användare