

Machine Learning - Kunskapskontroll 2

I denna kunskapskontroll ska du besvara nedanstående teoretiska frågor samt modellera MNIST. Notera att det alltså är två delar som ska genomföras.

Uppgiften görs och lämnas in individuellt men ni uppmuntras till att samarbeta och diskutera med varandra så länge som det ni lämnar in är ert eget arbete, precis som i arbetslivet. För de teoretiska frågorna nedan får ni inte använda ChatGPT (eller liknande).

På denna kunskapskontroll kan man få betygen IG / G / VG. För att bli godkänd så skall du visa att du använder koncepten vi lärt oss korrekt. Att uppnå godkänt är i sig inte ”enkelt” och kräver ansträngning. Satsar du på VG så förbered dig på att det blir mer arbete och kräver djupare förståelse samt problemlösning (se exakta betygskriterier i studiehandledningen). De som satser på VG behöver genomföra arbetet med att svara på de teoretiska frågorna samt modellera MNIST på ett bra sätt, men även bygga en applikation för att prediktera ”ny data”.

Börja så snart som möjligt för att snabbt upptäcka vad som kommer att vara utmanande samt kommunicera med varandra för att få tips.

Du ska lämna in kod, ett dokument med svar på de teoretiska frågorna samt göra en muntlig presentation av ditt arbete under en lektion. Om du inte kan delta vid lektionstillfälle för att göra en muntlig presentation kommer du att behöva spela in dig själv när du presenterar. Max 15 minuter.

Du ska även genomföra en självutvärdering där du besvarar följande tre frågor:

1. Har något varit utmanande i kursen/kunskapskontrollerna? Om ja, hur har du hanterat det?
2. Vilket betyg anser du att du ska ha och varför?
3. Något du vill lyfta fram till Terese?

Ladda upp dina lösningar i en mapp (repository) på GitHub. Glöm inte att göra mappen ”public” så att jag kan komma åt filerna. Lämna in en länk på itslearning.

Tid för muntlig presentation: Torsdag 26 feb 2026.

Deadline inlämning: Fredag 27 feb 2026 kl. 23.59.

Teoretiska frågor

Svara gärna kort och koncist.

1. All data i Python representeras av objekt. I Python består varje objekt av tre grundläggande delar. Vilka är dessa delar? Beskriv varje del kort.
2. Förklara skillnaden mellan mutabla och immutabla objekt i Python och ge exempel på varje.
3. Vad är ett set i Python? Vilka egenskaper särskiljer det från listor? Ge ett praktiskt exempel där det är mer effektivt att använda ett set än en lista.
4. Förklara vad en loop är och ge exempel på när den används.
5. Vad är en klass och hur skiljer sig en instans från själva klassen?
6. Vad är en funktion och varför använder programmerare dem? Förklara skillnaden mellan en funktion med returnvärde och en som inte returnerar något.
7. Förklara begreppet parameter jämfört med argument Python.
8. Titta på följande video om R: [R Programming](#). Vad är skillnaden mellan R och Python?
9. Kalle ska bygga en ML-modell och delar upp sin data i ”Träning”, ”Validering” och ”Test”, vad används respektive del för?
10. Julia delar upp sin data i träning och test. På träningsdataen så tränar hon tre modeller; ”Linjär Regression”, ”Lasso regression” och en ”Random Forest modell”. Hur skall hon välja vilken av de tre modellerna hon skall fortsätta använda när hon inte skapat ett explicit ”valideringsdataset”?
11. Vad är ”regressionsproblem”? Kan du ge några exempel på modeller som används och potentiella tillämpningsområden?
12. Hur kan du tolka RMSE och vad används det till.
13. Vad är ”klassificeringsproblem”? Kan du ge några exempel på modeller som används och potentiella tillämpningsområden?
14. Vad är en ”Confusion Matrix”?
15. Vad är K-means modellen för något? Ge ett exempel på vad det kan tillämpas på.
16. Förklara (gärna med ett exempel): Ordinal encoding, one-hot encoding, dummy variable encoding.
17. 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?
18. Vad är skillnaden mellan parametrar och hyperparametrar i en maskininlärningsmodell? Ge ett exempel på varje och förklara varför de inte kan optimeras på samma sätt.
19. Förklara skillnaden mellan overfitting och underfitting i en maskininlärningsmodell. Beskriv även hur man kan upptäcka respektive åtgärda dem.

Modellera MNIST

MNIST är ett standarddataset inom maskininlärning som används för att träna och testa algoritmer för bildigenkänning. Det består av 70 000 gråskalebilder av handskrivna siffror (0–9), där varje bild är 28×28 pixlar.

Del 1 – Skapa en ML-modell (G)

Använd maskininlärning för att modellera MNIST-datan. Du skall utvärdera minst två olika modeller i ditt arbete och göra ett komplett ML-flöde, från början där du laddar in data till slut där du utvärderar den bäst valda modellen på din testdata.

Hur du laddar ned MNIST-datan kan du se här:

```
1 import numpy as np
2 from sklearn.datasets import fetch_openml

1 mnist = fetch_openml("mnist_784", version = 1, cache = True, as_frame = False)
2 print(mnist.DESCR)

1 X = mnist["data"]
2 y = mnist["target"]
```

Del 2 - Skapa en Streamlit-applikation (VG)

De som satser på VG behöver genomföra arbetet med att svara på de teoretiska frågorna samt modellera MNIST på ett bra sätt (se betygskriterier i studiehandledningen för exakta detaljer), men även bygga en applikation för att prediktera ”ny data”.

- Kolla följande video om Streamlit: <https://www.youtube.com/watch?v=ggDaRzPP7A&list=PLgzaMbMPEHEx9Als3F3sKKXexWnyEKH45&index=12>
- Spara modeller med joblib: Se sidan 61 i kursboken ”Lär dig AI från grunden”.

Skapa en Streamlit-applikation där man med hjälp av en modell tränad på MNIST (kan vara den modell som du arbetat med i del 1) kan prediktera ”ny data” som matas in via Streamlit. Den ”nya datan” kan t.ex. komma från att användaren laddar upp en egen hand-ritad bild, att man ritar en siffra via Streamlit eller att man tar en bild via kameran på datorn. Du väljer själv.

Notera, generellt sett är utmaningen att man behöver transformera bilderna som tas innan de används för att skapa en prediktion.

Exempel för inspiration: [Lägg upp | LinkedIn](#)