MNIST-klassificering med maskininlärning

William Kabpimai

EC Utbildning

Kunskapskontroll 2 - Rapport

2025/03

Abstract

This project explores the application of various machine learning models to the MNIST dataset, which focuses on handwritten digit classification. The performance of Random Forest, Extra Trees, and Linear Support Vector Machine (SVM) models, as well as a Voting Classifier combining these, will be tested, compared and evaluated. The evaluation will be based on accuracy and detailed classification reports and reveal the effectiveness of ensemble methods, such as the Voting Classifier.

Förkortningar och Begrepp

RMSE = Root Mean Squared Error

SVM = Support Vector Machines

MNIST = Modified National Institute of Standards and Technology

ML = Machine Learning

Innehållsförteckning

1 Inledning

Maskininlärning kan användas för att klassificera bilder, exempelvis handskrivna siffror. För att testa och träna modellerna så används oftast datasetet MNIST, som då innehåller bilder på handskrivna siffror. I projektet undersöker vi hur de olika maskininlärningsmodellerna klassificerar MNIST datasetets siffror. Vi kommer att jämföra och utvärdera modellerna Random Forest, Extra Trees, SVM och Voting Classifier baserat på deras prestanda att korrekt klassificera de siffrorna i datasetet.

1.1 Syfte och frågeställning

Syftet är att utvärdera och jämföra prestandan för de olika maskininlärningsmodeller: Random Forest, Extra Trees, SVM och Voting Classifier på MNIST datasetet. Genom att analysera noggrannheten och klassificeringsrapporterna så kan vi identifiera den mest effektiva modellen för klassificering av handskrivna siffror.

Frågeställningen som arbetet besvarar är: Vilken av de undersökta maskininlärningsmodellerna har den högsta prestandan i klassificeringen av handskrivna siffror från MNIST datasetet?

2 Teori

2.1 Introduktion till maskininlärning

Maskininlärning är ett sätt att få datorer att lära sig saker på egen hand utan att vi behöver tala om exakt hur de ska göra. Istället för att ge datorn detaljerade instruktioner så matar vi in data och låter datorn hitta mönster och samband. På så sätt kan datorn lära sig att göra förutsägelser eller beslut. Detta kan vara användbart på många områden. Till exempel kan datorer lära sig att känna igen bilder eller tal.

2.2 Random Forest

Random Forest är en algoritm som används för att göra förutsägelser, till exempel att kategorisera objekt eller förutsäga numeriska värden. Istället för att bara använda ett träd, som kan göra fel, använder den en hel skog av träd. Varje träd får titta på lite olika delar av datan och sedan fråga några frågor om den. Vid en förutsägelse om ett objekt, får alla träd uttala sig. Datorn väljer därefter den åsikt som flest träd tycker är riktigt.

2.3 Extra Trees

Extra Trees är en annan algoritm som liknar Random Forest. Den använder också en skog av beslutsträd för att göra förutsägelser. Till skillnad från Random Forest där varje träd får titta på en slumpmässig del av datan så får varje träd i Extra Trees titta på alla exempel, vilket gör algoritmen ännu mer slumpmässig.

2.4 SVM

SVM är en form av algoritm som används för att dela upp data i olika grupper eller klasser. Den försöker hitta den bästa gränsen som skiljer grupperna åt. I enkelhet så är det en algoritm för att dela upp data i grupper. Rent tekniskt så försöker SVM hitta den gräns som har så stort avstånd som möjligt till de närmaste datapunkterna från varje grupp.

2.5 Voting Classifier

Voting Classifier är en teknik inom maskininlärning som kombinerar förutsägelser från flera olika modeller för att göra en slutgiltig förutsägelse. Istället för att förlita sig på en modell så låter Voting Classifier flera modeller rösta om vilket svar som är bäst. Rent praktiskt så kombineras förutsägelsen från modellerna för att ge mer träffsäkerhet i slutresultatet.

2.6 MNIST

MNIST är en stor dataset med handskrivna siffror. Datasetet innehåller 70 000 bilder av handskrivna siffror från 0 till 9 där varje bild är i gråskala och har dimensionerna 28x28 pixlar.

MNIST är en modifierad version av dataset som ursprungligen samlades in av National Institute of Standards and Technology (NIST).

2.7 Utvärderingsmetoder

När vi har tränat en maskininlärningsmodell är det viktigt att veta hur bra den är på att göra förutsägelser. Dessa utvärderingsmetoder är några av de få som finns.

2.7.1 Noggrannhet (Accuracy)

Noggrannhet är en mått för att utvärdera klassificeringsmodeller. Den mäter hur stor andel av förutsägelserna som är korrekta. Ifall modellen gissar rätt på 9 av 10 bilder så är noggrannheten 90% men den brukas ofta visas i decimalform, alltså 0,9.

2.7.2 Konfusionsmatris (Confusion Matrix)

En konfusionsmatris visar hur bra modellen är på att klassificera olika klasser. Med betoning på visar så är det en matris som visualiserar hur många gånger modellen gissade rätt och fel för varje klass.

2.7.3 Klassificeringsrapport

Klassificeringsrapporten ger en detaljerad bild av modellens prestanda. Den innehåller mått som precision, recall och F1-score för varje klass.

- Precision mäter hur träffsäker modellen är.
- Recall mäter hur många av de faktiska klasserna som modellen lyckades hitta.
- F1-score är ett kombinerat mått som tar hänsyn till både precision och recall.

3 Metod

3.1 Datainsamling och förberedelse

Följande bibliotek importerades in: Matplotlib för visualisering, scikit-learn för datainläsning, modellträning och utvärdering, samt Seaborn för grafisk visualisering av resultaten.

Från scikit-learn biblioteket importerades modellerna RandomForestClassifier, ExtraTreesClassifier, LinearSVC (SVM) och VotingClassifier. För att utvärdera modellernas prestanda importerades funktionerna accuracy_score, confusion_matrix och classification_report.

Från scikit-learn importerades funktionen fetch_openml som laddar in MNIST datasetet. Funktionen train_test_split delar sedan upp datan i 3 olika delar, träningsset, valideringsset och testset. 30 000 för träning, 30 000 för validering och resterande 10 000 för test som hålls undan för att simulera modellerna för ny och okänd data. Summan på hela datasetet blir då 70 000.

Eftersom vissa delar av koden använder slumpmässighet så skapas instansen np.random.seed(42) för att säkerställa att vi får samma resultat varje gång, eftersom detta är viktig för reproducerbarhet.

Bilddatan omvandlades till en Numpy array format, för kompabilitet med de maskininlärningsverktyg som användes.

3.2 Modellträning

För att träna modellerna på att känna igen handskrivna siffror från MNIST datasetet så användes träningsdatan, som bestod av 30 000 bilder. Modellerna RandomForestClassifier, ExtraTreesClassifier, LinearSVC och VotingClassifier tränades. Varje modell skapades med standardinställningar från scikit-learn och tränades individuellt på träningsdatan. Under träningen anpassade modellerna sina interna inställningar för att minska antalet felaktiga gissningar.

- För RandomForestClassifier och ExtraTreesClassifier tränades flera beslutsträd och deras gissningar kombinerades.
- För LinearSVC försökte modellen hitta den bästa linjen för att dela upp siffrorna i olika grupper.
- För VotingClassifier kombinerades gissningarna från de tre tidigare modellerna.

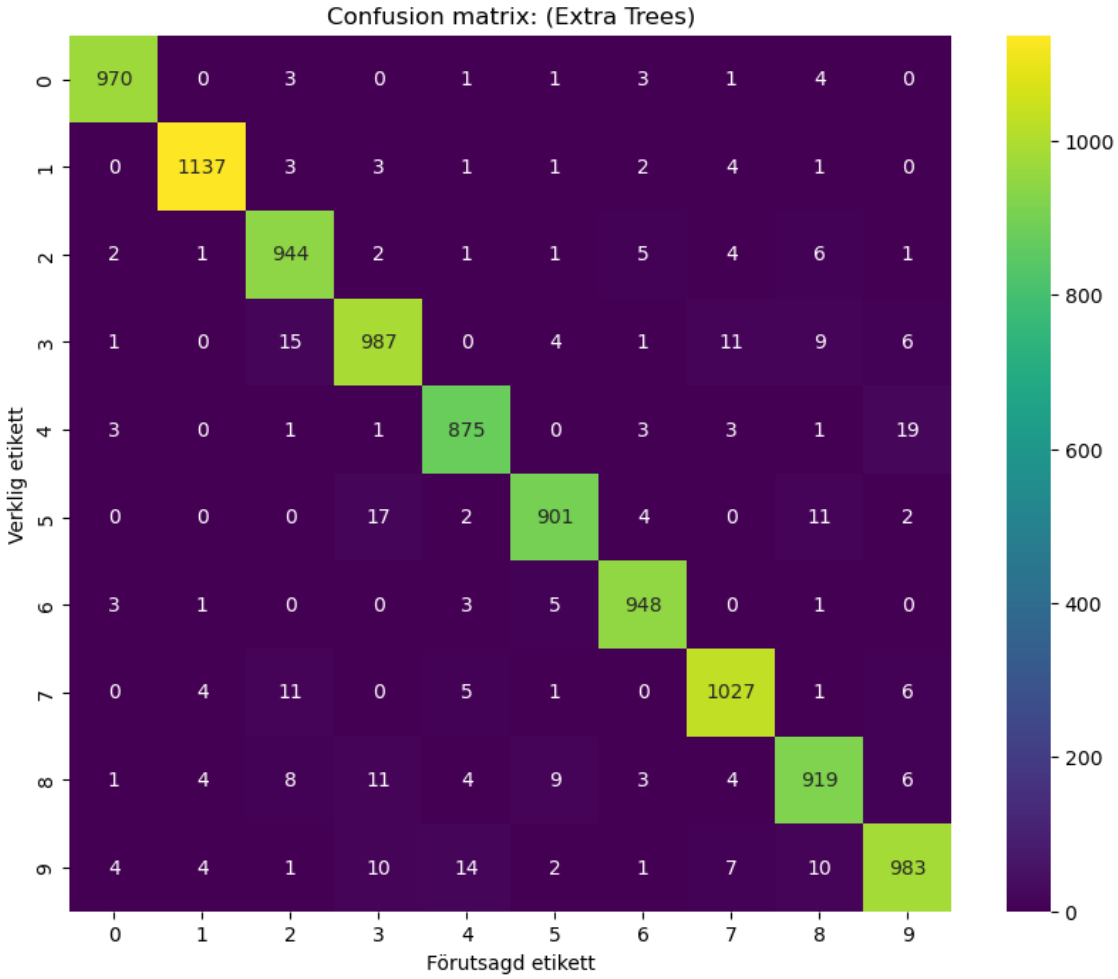
3.3 Modellutvärdering

För att utvärdera modellernas förmåga att känna igen handskrivna siffror användes testdatan, som bestod av 10 000 bilder. Modellernas prestanda mättes med hjälp av tre olika metoder. Först användes varje modell för att göra gissningar på testdatan. Sedan beräknades hur många av dessa gissningar som var korrekta med hjälp av funktionen accuracy_score. Därefter skapades en konfusionsmatris som visar hur många gånger varje siffra blev korrekt eller felaktigt gissad. Slutligen skapades en rapport som visade precision, återkallelse och F1-poäng för varje klass.

4 Resultat

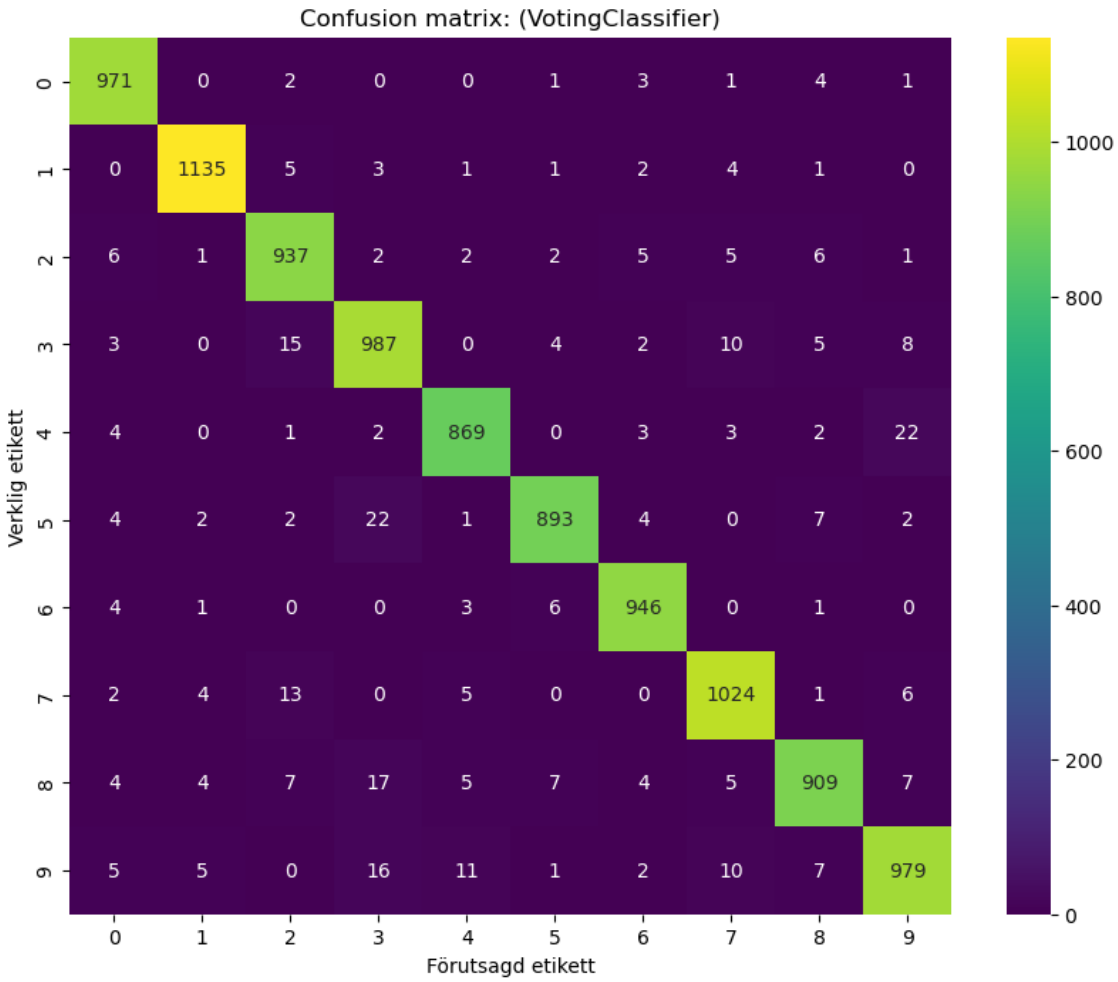
Noggrannhet för olika modeller i %	
Extra Tree	96,91 %
Voting Classifier	96,5 %
Random Forest	96,45 %
LinearSVC (SVM)	85,66 %

Matris 1: Konfusionsmatris för Extra Trees modell.



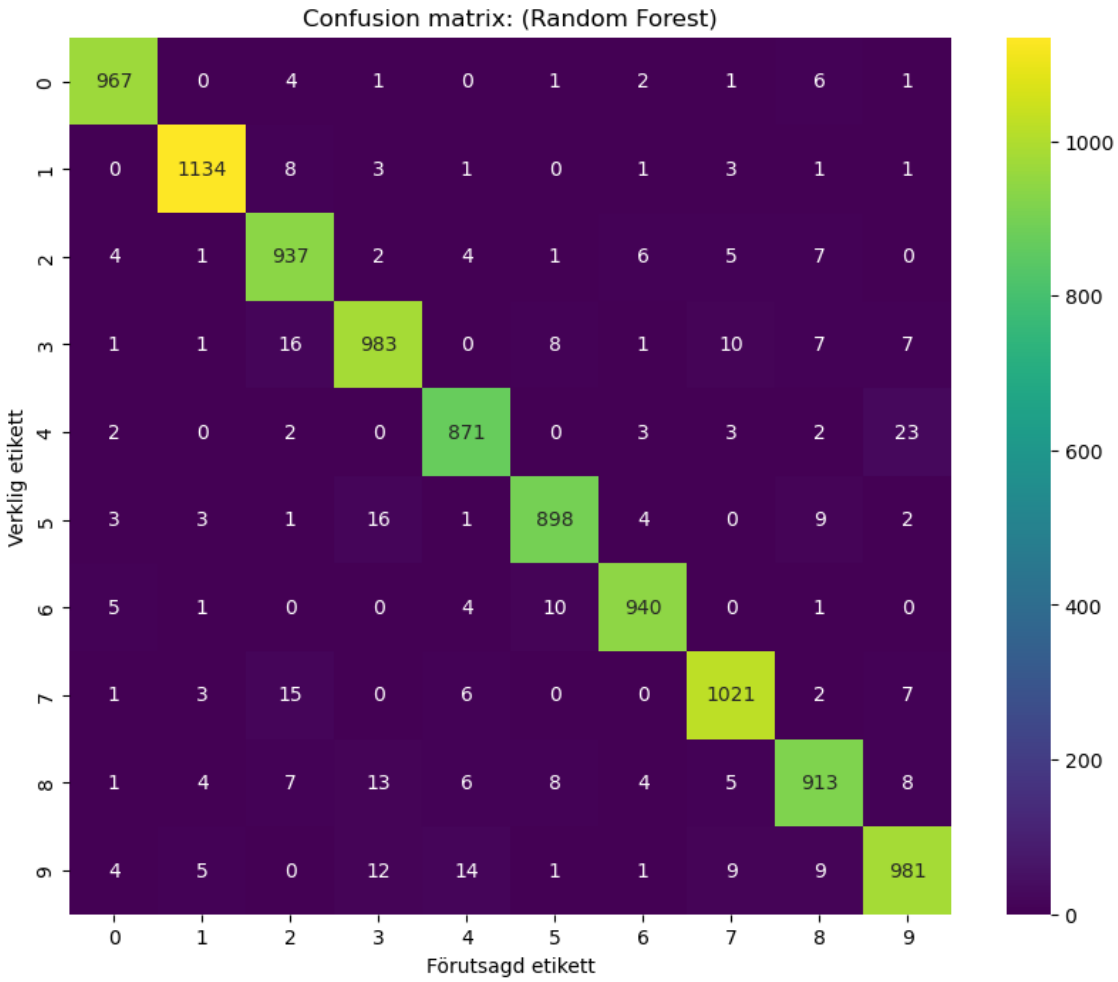
- Extra Trees modellen verkade ha 19 missar som högst på just 4 då den gissat 9 istället. Kanske var talet skriven med en stängd hatt. Det verkar också sticka ut då modellen har misstagit 9, 8 och 5 för nummer 3.

Matris 1: Konfusionsmatris för Voting Classifier modell.



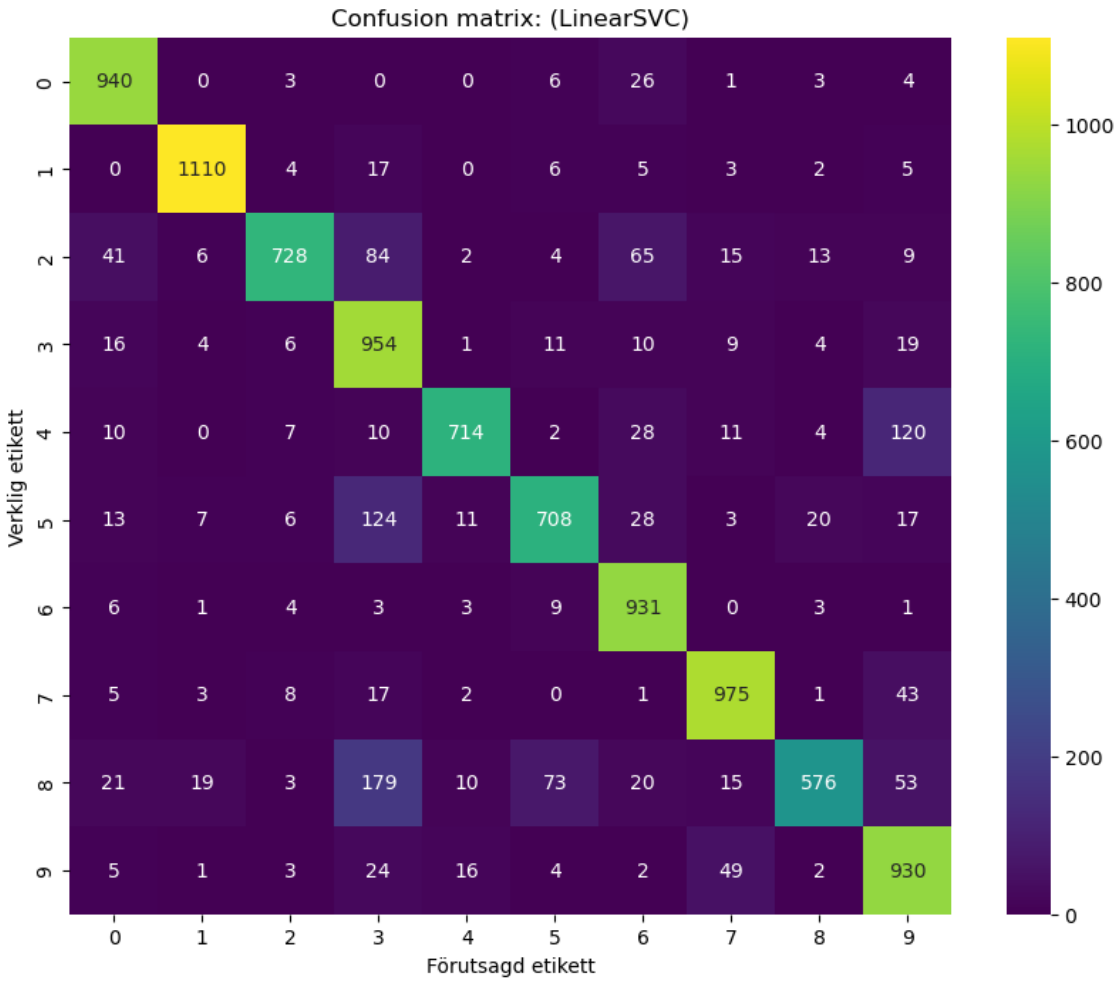
- Voting Classifier modellen verkar ha liknande svårigheter med samma tal som i Extra Trees modellen.

Matris 2: Konfusionsmatris för Random Forest modell.



- Random forest modellen verkade ha 23 missar som högst då modellen gissat 9 när det egentligen var en 4. Det kan vara så att 4 var skriven med en stängd hatt. Nummer 3 verkar också sticka ut då modellen har misstagit 9, 8 och 5 för nummer 3.

Matris 3: Konfusionsmatris för LinearSVC (SVM) modell.



- SVM modellen har presterat mycket sämre då gissningarna på nummer 3 och 9 sticker ut mest.

5 Slutsatser

I detta arbete har prestandan hos maskininlärningsmodellerna Random Forest, Extra Trees, SVM och en Voting Classifier utvärderats för klassificering av handskrivna siffror från MNIST datasetets testset.

Rent objektivt så uppnådde modell Extra Trees den högsta noggrannheten på testsetet, vilket innehåller data som modellen inte har tränats på. Resultaten visar att modellen har en bra prestanda i denna specifika utvärdering på ny okänd data.

Det är viktigt att notera att resultaten för Extra Trees och Voting Classifier var väldigt nära varandra. Extra Trees uppnådde en noggrannhet på 0,9691, medan Voting Classifier uppnådde 0,965. Det är möjligt att den lilla skillnaden betyder att modellernas prestanda kan variera beroende på den specifika datamängden.

Det är därför möjligt att resultaten skulle kunna se annorlunda ut om vi använde andra datamängder eller andra sätt att mäta hur bra modellerna är. Detta visar att resultaten är väldigt nära varandra och kan variera beroende på den datamängd som används.

6 Teoretiska frågor

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

- Träningsdatan används för att träna maskininlärningsmodellen genom mönster och relationer.

- Valideringsdatan används för att utvärdera modellens prestanda under träningen.

- Testdatan används efter att modellen är färdigtränad och under hela träningsprocessen ska testdatan vara okänd för att utvärdera modellens prestanda för ny data.

2. Julia delar upp sin data i träning och test. På träningsdatan 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”?

- Julia kan använda sig av korsvalidering för att utvärdera prestandan på sina modeller och senare behöver hon använda ett lämpligt utvärderingsmått.

3. Vad är ”regressionsproblem? Kan du ge några exempel på modeller som används och potentiella tillämpningsområden?

- Ett regressionsproblem handlar om att förutsäga ett värde som exempelvis huspriser eller temperaturen för imorgon. Några exempel på modeller är, Linjär regression, Polynominal regression och Random forest regression.

4. Hur kan du tolka RMSE och vad används det till?

$$RMSE = \sqrt{\frac{1}{n}\sum_{1}^{n}(y_i - \hat{y_i})^2}$$

- RMSE är ett felmått som visar hur långt ifrån en modells gissningar avviker sig från de riktiga värdena. Om man har en modell som förutsäger huspriser och RMSE är 1000, så innebär det att modellens gissningar avviker sig med 1000 från de riktiga värdena, i den valutan det nu handlar om.

5. Vad är ”klassificieringsproblem? Kan du ge några exempel på modeller som används och potentiella tillämpningsområden? Vad är en ”Confusion Matrix”?

- En klassificieringsproblem handlar om att sätta saker i olika kategorier som exempelvis hund eller katt. Tillämpningsområdena kan exempelvis vara att sortera spammejl, reklam eller vanlig mejl. Några exempel på modeller är, Logistisk regression, Decision trees och SVM.

- Confusion matrix är en tabell och används för att jämföra de faktiska klasserna med vad modellerna har gissat på. Det är ett verktyg för att utvärdera prestandan på klassificeringsmodellerna. Matrisen är ett sett för oss att förstå vad modellen gör rätt och fel.

6. Vad är K-means modellen för något? Ge ett exempel på vad det kan tillämpas på.

- K-means är en metod för att hitta grupper i data genom likheter. Det är en metod som oftast används inom oövervakad maskininlärning. K-means kan användas för att dela upp affärskunder i olika grupper, baserat på deras köptrender.

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

- Ordinal encoding används för data som kan rangordnas. Exempelvis kundomdöme, från dålig, sådär till bra. Med Ordinal encoding kan vi tilldela ett numeriskt värde som representerar ordningen i kundomdömet från 1 till 3 (dåligt till bra).

- One hot encoding används för data som inte har en tydlig ordning. Exempelvis färger, röd, grön och blå. Med One hot encoding så omvandlas kategorierna till binära tal i form av en vektor, alltså skulle det se ut: [0, 1]. Där röd = [1, 0, 0], grön = [0, 1, 0] och blå = [0, 0, 1].

- Dummy variable encoding liknar One hot encoding. Om vi jämför med tidigare färgexempel så tar Dummy encoding i praktiken bort en kolumn. Vektorerna skulle isåfall se ut: [1, 0], [0, 1], [0, 0] för respektive färg. Dummy encoding används då det finns faktorer i datan som starkt relaterar varandra, det finns olika nyanser av en färg. Eftersom relationerna är så starka så kan det förvirra medellen och göra dess gissningar osäkra.

8. 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 Julias påstående är valid. Göran ger en grundläggande definition på data inom statistik. Kategorisk data kan vara antigen nominal eller ordinal. Göran menar att en viss kategorisk data har någon inbördes ordning som exempelvis betyg eller storlek på kläder. Och när data inte har någon ordning så kan det exempelvis vara färger eller djur, vilket gör hans påstående korrekt.

- Det Julia påpekar är att data kan tolkas olika beroende på sammanhanget. Julia ger ett viktigt perspektiv där nominal data, alltså färger kan vara ordinal i ett sammanhang där exempelvis en viss färg är mer värderat, detta gör alltså att hennes påstående är också korrekt.

9. Vad är Streamlit för något och vad kan det användas till?

- Streamlit är en open source Python bibliotek för att skapa och dela webbappar för maskininlärning och datavetenskap. Streamlit gör det enklare att skapa webbappar med färre rader av kod, utan att behöva mycket kunskap inom webbutveckling. Det är ett verktyg för att enkelt kunna skapa en interaktiv webbapp för datavisualisering.

7 Självutvärdering

1. Utmaningar du haft under arbetet samt hur du hanterat dem.
   - Kod delen skulle jag säga var mer utmanande, det hjälpte att MNIST är som “hello world” fast för ML, jag skulle säga att koda samtidigt och kolla på youtube hjälpte enormt, lite som “code along” och jag känner att det funkar lite bättre för mig än när jag försöker göra det på lektionerna.
   - Jag försöker lära mig “the fundamentals” ifall jag stöter på något jag inte förstår och tar mig därifrån.

2. Vilket betyg du anser att du skall ha och varför.
   - Jag tror jag klara mig, förstår lite grundläggande om vissa koncept, lite sämre känsla på den tekniska delen och koden såklart, men det kommer.
   - Jag missade att normalisera pixelvärdena som kunde ha hjälpt träna modellerna, så jag tar med det till nästa gång.

3. Något du vill lyfta fram till Antonio?
   - Kul uppgift och väldigt tidskrävande.

Källförteckning

Datakälla:

LeCun, Y., Cortes, C. & Burges, C.J.C., 1998. *The MNIST database of handwritten digits*. http://yann.lecun.com/exdb/mnist/