Kunskapskontroll 2

Machine Learning – MNIST dataset



Camilla Månsson

EC Utbildning

Kunskapskontroll 2 – Machine Learning

Mars 2024

# Abstract

This study focuses on exploring and using the complete machine learning workflow with the MNIST dataset. The aim is to develop and train a model capable of accurately classifying handwritten digits. This task is accomplished by using three different models, SVM, Random Forest and KNN. The SVM model emerged as the top performer. The rather impressive accuracy of the SVM model shows its strength in dealing with complex classification tasks. This shows how machine learning can effectively recognize handwritten characters and be applied in various real-world scenarios.

Innehållsförteckning

[Abstract 2](#_Toc162037645)

[1 Inledning 1](#_Toc162037646)

[1.1 Syfte 1](#_Toc162037647)

[1.2 Frågeställning 1](#_Toc162037648)

[2 Teori 2](#_Toc162037649)

[2.1 Klassifikationsmodeller 2](#_Toc162037650)

[2.1.1 SVM 2](#_Toc162037651)

[2.1.2 Random Forest 2](#_Toc162037652)

[2.1.3 KNN 2](#_Toc162037653)

[2.2 GridSearch 2](#_Toc162037654)

[2.3 Metrics 2](#_Toc162037655)

[2.3.1 Accuracy 2](#_Toc162037656)

[2.3.2 Precision 3](#_Toc162037657)

[2.3.3 Recall 3](#_Toc162037658)

[2.3.4 F1 3](#_Toc162037659)

[2.3.5 Classification report 3](#_Toc162037660)

[2.3.6 Confusion Matrix 3](#_Toc162037661)

[3 Metod 4](#_Toc162037662)

[3.1 Verktyg 4](#_Toc162037663)

[3.2 Dataset 4](#_Toc162037664)

[3.3 Explorativ dataanalys 4](#_Toc162037665)

[3.4 Förberedelse av data 4](#_Toc162037666)

[3.5 Val av modell 4](#_Toc162037667)

[3.6 Modellträning och utvärdering 5](#_Toc162037668)

[4 Resultat och Diskussion 6](#_Toc162037669)

[4.1 Modellernas resultat 6](#_Toc162037670)

[4.2 Den slutgiltigt valda modellens resultat 7](#_Toc162037671)

[5 Slutsatser 8](#_Toc162037672)

[6 Teoretiska frågor 9](#_Toc162037673)

[7 Självutvärdering 12](#_Toc162037674)

[Källförteckning 13](#_Toc162037675)

# Inledning

I den moderna digitala eran, där data genereras i en omfattning som överstiger vår föreställningsförmåga, har maskininlärning blivit en viktig och nödvändig teknologi. Data samlas kontinuerligt in från en mängd olika källor - från användares beteende på internet till sensorer i smarta enheter för att bara nämna några. Maskininlärning, genom sin förmåga att analysera stora datamängder, upptäcka mönster och göra prediktioner, har spelat en avgörande roll i flera teknologiska innovationer som påverkar våra liv på många sätt.

Ett välstuderat exempel på maskininlärningens styrkor är dess förmåga att klassificera handskrivna siffror med hög precision. Avancerade algoritmer gör det möjligt för datorer att noggrant särskilja olika handskrivna siffror. I denna rapport undersöks processen för maskininlärningsmodellering på ett av de mest kända och flitigt använda dataseten inom området - MNIST. Detta dataset innehåller hela 70 000 exemplar av handskrivna siffror.

## Syfte

Syftet med denna rapport är att undersöka vilken, av tre valda modeller, maskininlärningsmodell som presterar bäst på MNIST dataset.

## Frågeställning

För att uppfylla syftet så kommer jag besvara följande frågeställning:  
  
1. Vilken av de valda modellerna kommer prediktera bäst accuracy score?  
2. Kommer justering av hyperparametrar innebära en förbättring av modellen?  
3. Kommer den bästa modellen att prediktera MNIST med över 95 % accuracy?

# Teori

Teoridelen av denna rapport syftar till att kort förklara några grundläggande begrepp som används i denna uppgift.

## Klassifikationsmodeller

En klassifikationsmodell är en typ av maskininlärningsalgoritm som används för att kategorisera data i olika klasser eller kategorier baserat på dess egenskaper eller attribut. Huvudsyftet med denna typ av modeller är att göra prediktioner eller fastställa vilken klass eller kategori en given datapunkt tillhör. Klassifikationsmodellen tränas med hjälp av märkt exempeldata, där den korrekta klassen för varje datapunkt är förbestämd. Genom att analysera och lära sig mönster från dessa exempel kan modellen sedan tillämpas för att göra prediktioner för nya, oidentifierade datapunkter.

### SVM

Support Vector Machine, SVM, är en mångsidig och kraftfull maskininlärningsmodell som är lämplig att använda vid klassificeringsproblem (Géron, 2019). Denna modell är en typ av linjär klasificerare som kan hantera både linjära och icke linjära separerbara dataset. SVM klassificering bygger på principen om "maximal marginalklassificering" vilket betyder att modellen försöker skapa största möjliga marginal mellan datapunkterna hos två olika klasser. För att lösa problemet vid icke linjär separerbar data använder SVM något som kallas för ”kernel-tricks”. Tack vare detta trick kan datan transformeras till en högre dimension och där separeras linjärt (Géron, 2019). SVM är en modell som passar bra för små och medelstora dataset. Det är dock en modell som är känslig för skalan på datan och därför standardiserar man sin data med hjälp av StandardScaler i Scikit-learn (Prgomet, 2024). För att hitta de bästa hyperparametrarna och därmed skapa den bästa modellen kan GridSearch användas.

### Random Forest

Random Forest är en maskininlärningsmodell som använder en teknik som kallas ensemble learning vilket innebär att den använder flera beslutsträd. I en Random Forest skapas flera beslutsträd, där varje träd tränas på olika delmängder av data med slumpmässigt urval av funktioner. Varje träd är unikt och när det är dags för prediktion sammanställs prediktionerna från varje träd för att avgöra det mest förekommande resultatet för klassificeringen. Till skillnad från SVM så är Random Forest inte så känslig för skalan på datan (Géron, 2019).

### KNN

KNN, förkortning för K-Nearest Neighbors, är en intuitiv maskininlärningsalgoritm som kan användas för klassificering. I KNN-algoritmen klassificeras eller predikteras en ny observation genom att jämföra det med de k närmsta observationerna(grannarna) i träningsdatan. Här representerar K antalet närmsta grannar som används för att fatta beslutet. Vid klassificering tilldelas den nya observationen den klass som är mest förekommande bland dess k närmsta grannar.

## GridSearch

I Scikit-learn finns en funktion som kallas GridSearch som hjälper till att hitta de hyperparametrar som gör att modellen predikterar bäst.

## Metrics

Inom maskininlärning är metrics olika mätbara värden som används för att bedöma vilken modell som presterar bäst eller är mest träffsäker.

### Accuracy

Accuracy score mäter andelen korrekta klassprediktioner.

### Precision

Precision visar andelen positiva prediktioner som faktiskt är korrekta.

TP = Number of true positive   
FN = Number of false positive  
(Progomet, 2024)

|  |  |  |
| --- | --- | --- |
|  | Predicted Negative | Predicted Positive |
| Actual Negative | True Negative (TN) | False Positive (FP) |
| Actual Positive | False Negative (FN) | True Positive (TP) |

Figur 1: Tabell för att visa TN, FP, FN och TP.

### Recall

Recall visar andelen av den positiva klassen som vi predikterar korrekt.

TP = Number of true positive  
FN = Number of false negative  
(Progomet, 2024)

Se även figur 1 ovan för bättre förståelse.

### F1

F1 Score är också en typ av metric och beror på precision och recall, se nedan, men det finns ingen direkt tolkning av det.

(Progomet, 2024)

### Classification report

I ScikitLearn finns något som kallas Classification report som tar fram samtliga ovan nämnda metrics och visar resultaten för varje klass.

### Confusion Matrix

En Confusion Matrix är ett verktyg som används för att utvärdera hur väl en klassificeringsmodell presterar. Den består av två dimensioner, en med det faktiska värdet samt en dimension med det predikterade värdet. Om vi använder vårt fall med datasetet MNIST blir det en kvadratisk matris där varje rad representerar de faktiska klasserna dvs 0-9. Varje kolumn representerar de predikterade klasserna från modellen. I varje cell i matrisen står antalet observationer. På så sätt kan man lätt se antalet observationer som modellen har predikerat rätt respektive fel.

# Metod

I detta avsnitt presenteras de metoder som användes för att modellera MNIST och att hitta den modell som presterar bäst.

## Verktyg

För att skriva koden för uppgiften har Python använts i Jupyter Notebook.

## Dataset

Det dataset som använts för uppgiften är alltså MNIST som består av 70 000 handskrivna siffror. Bilderna är svart-vita och föreställer siffrorna mellan 0-9. Alla bilderna är har en storlek på 28x28 pixlar. För att erhålla MNIST databas användes metoden 'fetch\_openml', vilket är ett verktyg för att hämta data från det öppna datalagret för maskininlärning.

## Explorativ dataanalys

En kort explorativ dataanalys (EDA) genomfördes för att få en förstålse för datan. Exempelvis extraherades några av de första siffrorna i datasetet för att ge en överblick över hur siffrorna ser ut.  
  
En bild som visar diagram, linje, Graf

Automatiskt genererad beskrivning

Figur 2: Tre av siffrorna i datasetet.

## Förberedelse av data

Med tanke på MNIST omfattning minskades datamängden för att effektivisera hela processen. I samma steg delades datan upp i träning, valdiering och test. Eftersom SVM är känslig för skala applicerades StandardScaler för att normalisera datan. Detta steg är kanske inte nödvändigt för alla modeller som sedan valdes.

## Val av modell

Uppgiften krävde jämförelse mellan minst två modeller. Eftersom det rörde sig om ett klassificeringsproblem valde jag att utforska olika klassificerare. För att vägledas i valet av modeller använde jag bla scikit-learns ”Choosing the right estimator”. Det är en bild som illustrerar ett beslutsträdsschema utformat för att hjälpa användare att välja den mest lämpliga maskininlärningsalgoritmen (estimator) baserat på det specifika problemets karaktär och datans egenskaper. Vidare läste jag också om MNIST på Wikipedia där fördelaktiga klassificerare för datasetet presenterades. Utifrån det valde jag att fokusera på tre olika modeller för klassificering: Support Vector Machine, Random Forest och K-Nearest Neighbor.

## Modellträning och utvärdering

Varje modell tränades separat på träningsdata och utvärderades med classification report samt confusion matrix på valideringsdata. GridSearch användes för att finjustera hyperparametrar och modellerna tränades om med de optimala hyperparametrarna. Därefter utfördes utvärdering på testdatan för varje modell. Den bäst presterande modellen valdes ut efter att processen upprepats för alla tre modeller. Efter detta hämtades hela datasetet och datan delades upp i tränings-, validerings- samt testuppsättningar. StandardScaler användes för att standardisera datan innan en SVM-klassificerare skapades med de bästa hyperparametrarna. Modellen evaluerades sedan på testdatan med classification report och confusion matrix innan den sparades med hjälp av joblib.

# Resultat och Diskussion

I detta avsnitt presenteras resultaten från modelleringen av MNIST-datasetet tillsammans med några korta diskussionspunkter.

## Modellernas resultat

I tabellen nedan presenteras resultatet efter träning av modellen på träningsdatan och utvärdering på valideringsdatan.

|  |  |
| --- | --- |
| **Modell** | **Accuracy Score** |
| SVM | 0.94 |
| Random Forest | 0.95 |
| KNN | 0.91 |

Nedan presenteras resultatet efter GridSearch och optimering av hyperparametrar. De bäst funna hyperparametrarna visas också.

|  |  |  |
| --- | --- | --- |
| **Modell** | **Hyperparametrar efter GridSearch** | **Accuracy Score** |
| SVM | 'C': 0.5, 'gamma': 1, 'kernel': 'poly' | 0.96 |
| Random Forest | 'max\_depth': 30, 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 300 | 0.95 |
| KNN | 'n\_neighbors': 4, 'weights': 'distance' | 0.91 |

Från resultatet går det att utläsa att SVM har en högre accuracy score vilket innebär att den presterar bäst. Detta kan tänkas bero på SVM-modellers förmåga att hantera högdimensionella data.

Att Random Forest och KNN inte visar en förbättring av accuracy score efter GridSearch skulle kunna bero på felaktiga parameterinställningar eller ha med modellens komplexitet att göra.

Det är värt att notera att användning av en mindre del av datamängden kan leda till missvisande eller felaktiga resultat samt att det finns bättre lämpade hyperparametrar. Trots detta bedömdes det som ett strategiskt beslut för att effektivisera processen med avsikten att genomföra en utvärdering på hela datasetets testdata. Tanken var att vid behov justera processen.

## Den slutgiltigt valda modellens resultat

Baserat på de presenterade resultaten valdes SVM-modellen för att tränas om med de bästa hyperparametrarna på hela träningsdatan efter att ha hämtat datasetet på nytt. Det resulterade i en accuracy score på 0.98. Nedan följer en bild som visar resultatet av Classification report och en bild över Confusion Matrix, båda baserade på den fulla testdatan.

En bild som visar text, skärmbild, nummer, Teckensnitt

Automatiskt genererad beskrivningEn bild som visar text, skärmbild, diagram, kvadrat

Automatiskt genererad beskrivning

När modellen får träna på en större mängd data har den möjlighet att få en bättre förståelse för variationen och komplexiteten i datan. Detta ledde förmodligen till en högre accuracy score. Användning av en större datamängd kan också minska risken för överanpassning.

Under processens gång övervägde jag om jag skulle utforska ytterligare justeringar av hyperparametrarna eller använda andra tillvägagångssätt. Jag hade tankar på om dimensionsreducering med PCA vore lämpligt. Emellertid bedömde jag att modellerna redan presterade tillräckligt väl för den aktuella uppgiften och valde att avstå från att fördjupa mig i de tankarna.

# Slutsatser

Återvändande till den ursprungliga frågeställningen ger följande insikter.

1. Vilken av de valda modellerna kommer prediktera bäst accuracy score?

Den modell som presterade bäst var Support Vector Machine.

1. Kommer justering av hyperparametrar innebära en förbättring av modellen?

För SVM-modellen visade justeringen av hyperparametrarna en förbättring av prediktionsförmågan.

1. Kommer den bästa modellen att prediktera MNIST med över 95 % accuracy?

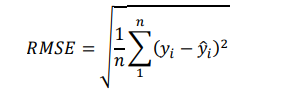
Ja, det gjorde den. Slutresultatet blev till och med 98% vilket tyder på att SVM-modellen har en stark förmåga att klassificera handskrivna siffror korrekt.  
  
Även om min SVM-modell visade en hög noggrannhet i sina prediktioner finns det klart utrymme för ytterligare förbättringar. Det är troligt att andra maskininlärningsmodeller, såsom neurala nätverk, kan uppnå ännu bättre resultat. Sammanfattningsvis visar resultaten att min SVM-modell presterade mycket väl samt demonstrerar dess förmåga och pålitlighet vid klassificering av handskrivna siffror i MNIST-datasetet.

# Teoretiska frågor

1. **Kalle delar upp sin data i ”Träning”, ”Validering” och ”Test”, vad används respektive del för?**   
   Den data som finns i träningsdelen används för att träna modellerna medan valideringsdelen används för att sedan kunna välja ut den bästa modellen. Med hjälp av den data som finns i test utvärderas den slutgiltiga modellen.
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 istället göra crossvalidation för att utvärdera vilken av modellerna som presterar bäst. Det innebär att träningsdatan delas in i flera mindre delar och på så sätt kan hon validera modellerna på ny data. Hon väljer sedan den modell som ger bäst resultat enligt det valda prestandamåttet. Vid utvärdering av regressionsmodeller kan man välja den modell som har lägst RMSE (Root Mean Squared Error).

1. **Vad är ”regressionsproblem? Kan du ge några exempel på modeller som används och potentiella tillämpningsområden?**   
   Regressionsproblem är ett typ av maskininlärningsproblem där vi modellerar och vill prediktera den beroende variabeln med kontinuerliga värden baserat på en eller flera oberoende variabler. Några exempel på modeller som används är Linjär Regression, Beslutsträd och Random Forest Regression. Potentiella tillämpningsområden är vid olika typer av ekonomiska prognoser som tex att prediktera lön baserat på ålder och utbildning eller huspriser baserat på läge och bostadsyta.
2. **Hur kan du tolka RMSE och vad används det till:**

  
RMSE (Root Mean Squared Error) kan vi tolka som våra prediktioners medelavstånd till de sanna värdena. Genom att kvadrera skillnaderna blir positiva och negativa avstånd likvärdiga, och när roten ur tas kan vi återigen relatera resultatet till ursprungliga enheter. RMSE är ett mått på hur träffsäker en modell är. Den modell som har lägst RMSE på osedd data är oftast den bästa modellen.

1. **Vad är ”klassificieringsproblem? Kan du ge några exempel på modeller som används och potentiella tillämpningsområden? Vad är en ”Confusion Matrix”?**   
   Klassificeringsproblem handlar om att försöka förutsäga vilken kategori eller klass en given ingångsdata tillhör. Det kan vara binära klassificeringsproblem, där det finns två möjliga klasser, som exempelvis Ja/Nej eller Man/Kvinna. Exempelvis kan vi försöka förutse om en kund kommer att tacka ja eller nej till ett rabatterbjudande, eller om en bild föreställer en man eller kvinna. Vid problem med flera klasser används multinomial klassificering. Logistic regression, K-Nearest Neighbors och Support Vector Machines är alla exempel på klassificeringsmodeller. Exempel på tillämpningsområde skulle kunna vara prediktering av blommor, spamfiltrering och bildigenkänning.

En Confusion Matrix är ett verktyg som används för att utvärdera hur väl en klassificeringsmodell presterar. Den består av två dimensioner, en med det faktiska värdet (det sanna värdet) samt en dimension med det predikterade värdet. Genom att jämföra varje prediktion som modellen gör med det faktiska värdet för den prediktionen ger Confusion Matrix en överblick över modellens prestanda.

1. **Vad är K-means modellen för något? Ge ett exempel på vad det kan tillämpas på.**I K-means modellen har datan inga ”labels” men med hjälp av en algoritm försöker dela in datan i olika grupper/kluster. Målet i K-means är alltså att dela in datan i grupper baserat på likheter mellan observationer. Varje observation tilldelas det kluster vars centroid (centrum) är närmst. Det är en generellt snabb och skalbar algoritm som passar bra till stora dataset. Ett exempel på tillämpningen av K-means är inom marknadssegmentering. Om vi har data som avser kunders köpbeteende kan vi med hjälp av K-mean dela in dem i grupper baserat på liknande beteendemönster vilket kan vara till hjälp för att rikta marknadsföringsstrategier mer effektivt.
2. **Förklara (gärna med ett exempel): Ordinal encoding, one-hot encoding, dummy variable encoding. Se mappen ”l8” på GitHub om du behöver repetition.**  
   I ordinal encoding får varje unik kategori ett värde i form av en siffra som baseras på dess rangordning. Tex om vi har en kategori kundnöjdhet med tre möjliga värden – ”missnöjd”, ”nöjd” och ”mycket nöjd” kan de kodas om som 1, 2 respektive 3.

I One-hot encoding representeras varje kategoriskt värde som en binär vektor där alla element är noll förutom en, som representerar indexet för det aktuella värdet. Till exempel, om vi har tre binära variabler, kan One-hot encoding se ut så här:

[['red']  
['green']  
['blue']]  
[[0. 0. 1.]  
[0. 1. 0.]  
[1. 0. 0.]]  
  
En dummy variable är en binär variabel som används för att representera olika kategorier eller grupper inom en kategorisk variabel. Istället för att skapa en binär variabel för varje kategori, skapas en variabel mindre än antalet kategorier.

1. **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?**

Julia har rätt. Det beror på kontexten. När vi pratar om färger som röd, grön och blå har de generellt sett ingen inbördes ordning och klassificeras därför som nominala. Men om vi säger att en röd skjorta är finast på festen, tolkar vi färgen som ordinal eftersom den har en rangordning i den specifika situationen.

1. **Kolla följande video om Streamlit: https://www.youtube.com/watch?v=ggDaRzPP7A&list=PLgzaMbMPEHEx9Als3F3sKKXexWnyEKH45&index=12 Och besvara följande fråga: - Vad är Streamlit för något och vad kan det användas till?**Stremlit är ett verktyg som kan användas för att skapa dataapplikationer genom python på ett enkelt sätt. Streamlit är tillgänglig för vem som helst att använda och undersöka och modifiera. Tex kan den användas för att presentera data på ett överskådligt sätt.

# Självutvärdering

1. Utmaningar du haft under arbetet samt hur du hanterat dem.   
   Att balansera studier med att ha små barn är en utmaning. Jag strävar efter att använda min tid så effektivt som möjligt. Den verkliga utmaningen har nog varit att verkligen förstå kursinnehållet och hur jag skulle närma mig den examinerande uppgiften. För att tackla dessa utmaningar har jag vänt mig till mina studiekompisar för hjälp och stöd.
2. Vilket betyg du anser att du skall ha och varför.   
   Jag tycker att jag har uppnått målen för godkänt. Jag anser att jag kan redogöra för det som krävs enligt kursens mål och kan implementera metodik för modellering.
3. Något du vill lyfta fram till Antonio?  
   Det har varit en utmanande men rolig kurs. Jag förstår att syftet med Streamlit-delen i den examinerande uppgiften är att lösa problem självständigt men jag hade gärna sett att den delen ingick lite mer i kursen.

# Källförteckning

Géron, A. (2019). *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems* 2nd ed. O’Reilly Media, Inc.

MNIST database. (March 2024). In Wikipedia. Retrieved from <https://en.wikipedia.org/wiki/MNIST_database>

Scikit-learn: Machine Learning in Python. (March 2024). Retrieved from  
<https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html>

Prgomet, A. (March 2024). Classification in Machine Learning [PowerPoint slides]. Retrieved from  
<https://github.com/AntonioPrgomet/machine_learning_ds23/blob/main/l2/02_klassificering.pptx>

Prgomet, A. (March 2024). Support Vector Machines [PowerPoint slides]. Retrieved from <https://github.com/AntonioPrgomet/machine_learning_ds23/blob/main/l5/07_svm.pptx>