Digit recognintion with Machine learning

Exploring classification problems and image processing in Python



Nike Sandberg Egbalewon

EC Utbildning

Kunskapskontroll 2

2024-03

# Abstract

Machine learning is a powerful tool to be used to solve complex problems. In this paper, three classification models have been used to test the accuracy of which images of hand written digits can be correctly predicted. As it shows, Support Vector Machines (SVM) is the model with the highest accuracy score, predicting the images correct 99,6 % of the time. To further test the model, it should be used on new images of hand written digits.

Innehållsförteckning

[Abstract ii](#_Toc162026184)

[1 Inledning 1](#_Toc162026185)

[2 Teori 2](#_Toc162026186)

[2.1 MNIST 2](#_Toc162026187)

[2.2 Klassfikationsmodeller 2](#_Toc162026188)

[2.2.1 Support Vector Machines (SVM) 2](#_Toc162026189)

[2.2.2 Logistisk regression 2](#_Toc162026190)

[2.3 MNIST 2](#_Toc162026191)

[3 Metod 3](#_Toc162026192)

[4 Resultat och Diskussion 4](#_Toc162026193)

[5 Slutsatser 5](#_Toc162026194)

[6 Teoretiska frågor 6](#_Toc162026195)

[7 Självutvärdering 8](#_Toc162026196)

[Källförteckning 9](#_Toc162026197)

# Inledning

Trots att maskininlärning och artificiell intelligens inte är förhållandevis nya praktiker och fenomen, har de, de senaste åren, tagit plats i samhällsdebatter, vetenskapliga framgångar och nyskapande möjligheter i vardagslivet på ett sätt och med en framfart som sällan skådats.

Maskininlärning kan enligt Géron sammanfattas som ”the science (and art) of programming computers so they can learn from data” (2019, s.2). Det första mötet med maskininlärning för gemene man var spam-filtret på 1990-talet som skapade möjligheter att detektera skräppost. Genom en modell som lärde sig vilka ord, fraser och mönster som skräppost ofta innehöll kunde den filtrera bort oönskad epost.

Med hjälp av maskininlärning kan vi därför hantera komplexa problem som annars med hjälp av traditionell programmering skulle kräva ett enormt manuellt arbete för att skriva uttömmande regler, algoritmer och utvärderingar som varken skulle bli tids- eller kostandseffektivt. Maskininlärning öppnar dörrar för effektiviseringar och problemlösningar i samhället, men även nya marknader och vetenskapliga upptäckter.

Digitaliseringen av tidigare analoga arbetssätt är en typ av effektivisering och förändring som präglat samhället de senaste åren. Handskrivna blanketter, signaturer och anteckningar lyser alltmer med sin frånvaro i kontakten med banker, myndigheter, företag och samhällsfunktioner i stort. I denna rapport utforskas tillämpningsområdet hur maskininlärning kan användas för att känna igen och korrekt prediktera mänsklig handskriven text.

Syftet med denna rapport är undersöka hur effektiva maskininlärningsmodellerna K nearest Neighbors, logistisk regression och Support Vector Machines är på MNIST-datasetet. För att uppfylla syftet så kommer följande frågeställning att besvaras:

1. Bland klassificeringsmodellerna K Nearest Neighbors, logistisk regression och Suport Vector Machines, vilken modell har bäst förmåga att korrekt prediktera handskrivna siffror?

# Teori

## MNIST

MNIST-dataset består av 70000 bilder av handskrivna siffror, där varje bild har klassificerats utifrån siffran den representerar. Datasetet är därmed användbart för att träna olika klassificeringsmodeller på. Géron, 2019, s. 85-86).

## Klassifikationsmodeller

Klassifikationsmodeller är ett annat exempel på maskininlärning som använder sig av väglett lärande. Som namnet antyder är det en modell som delar in den beroende variabeln i ”klasser”. MNIST-datasetet är ett gott exempel på ett dataset som kan tillämpas med hjälp av klassifikationsmodeller. Varje bild i datasetet har klassificerats med en siffra från 0-9, och vi kan använda klassificeringsmodeller för att prediktera vilken siffra varje bild innehåller. (Prgomet, 2023)

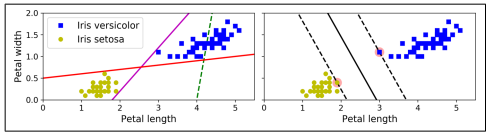
### Support Vector Machines (SVM)

Support vector machines är en modell för både klassificerings- och regressionsproblem. Modellen drar en linje för att separera de olika klasserna. Målet med denna linje är att göra den så bred som möjligt. Med det menar vi att det är ett så stort avstånd som möjligt mellan datapunkterna i de olika klasserna. Instanserna som ligger precis vid vägens gräns kallas för support vectors och är inringade i figur 1. (Géron, 2019, s. 153-154)

Vi vill ha så få ”margin violations” som möjligt, det vill säga instanser som befinner sig över marginalerna och ”på” vägen. Hur bred vägen är och hur många margin violations vi tillåter styrs av modellens hyperparameter C. Ett lågt C skapar en bredare väg, men tillåter fler margin violations medan ett högre C skapar en smalare väg men färre margin violations. Det optimala värdet på C för modellen kan väljas med hjälp av GridSearchCV i Scikit learn. (Prgomet, 2023)

#### Feature scaling

Att skala variablerna, även kallat ”Feature scaling”, är ett viktigt steg för att säkerställa den bredaste vägen möjligt då SVM-modellen är känslig för hur variabler är skalade. StandardScaler i Scikit-Learn kan användas för detta. (Géron, 2019, s. 154)



Figur 1.Linjär Support vector Machine-modell

#### Ickelinjär SVM

För ickelinjära klassificieringsplroblem finns det olika tekniker som kan användas med SVM-modeller. En matematisk teknik som heter kernel trick gör det möjligt att få samma resultat som om vi tillsatt flera polynominala variabler, utan att faktiskt tillsätta dem. Detta fungerar även på Gaussian Radial Basis Function (RBF) där kernel trick gör det möjligt att få samma resultat som om vi tillsatt ”similarity features” utan att faktiskt göra det. Återigen är GridSearchCV ett hjälpsamt verktyg för att finna den mest optimala tekniken och hyperparametrarna. (Géron, 2019, s. 157-160).

### Logistisk regression

Logistisk regression hanterar, namnet till trots, klassificeringsproblem. Den logistiska regressionsmodellen är en binär klassificeringsmodell som genom att uppskatta sannolikheten att indatan tillhör en viss klass predikterar vilken klass den tillhör. Är sannolikheten större än 50 % att instansen tillhör klassen predikterar den ja (1), annars nej (0). (Géron, 2019, s. 142)

### K Nearest Neighbors (KNN)

K Nearest Neighbor kan användas till både regressions- och klassificationsproblem. Det använder sig av icke-generaliserande lärande och ”förvarar” endast instanserna i datasetet med deras korresponderande y-värden. För att prediktera ny data räknar modellen ut avståndet mellan den nya instansen och alla andra i datasetet och räknar ut k antal närmsta ”neighbors”. Modellen tillskriven den nya instansen en klass utifrån vilken klass som är mest förekommande bland de k närmsta neighbors. Valet av optimala värdet på parametern k kan finnas genom GridSearchCV. (Scikit-Learn)

### Confusion Matrix

En confusion matrix är ett användbart verktyg vid hanteringen av klassificeringsproblem. Den kan användas för att utvärdera klassificeringsmodeller. Det är en tabell med två dimensioner, det sanna värdet och det predikterade värdet. Tabellen visar en fördelning över hur modellen har predikterat de olika instanserna i datasetet och hur dessa predikteringar förhåller sig till det sanna värdet. (Prgomet, 2023)

# Metod

Alla nödvändiga biblioteket i Python importerades som ett första steg i maskininlärningsprojektet. Därefter laddades MNIST-datan in och sparades i en X och y-variabel, för att sedan delas in i tränings- och testset. De första 60000 instanserna delades in i träningssetet och återstående 10000 instanserna i testsetet, se figur 2.

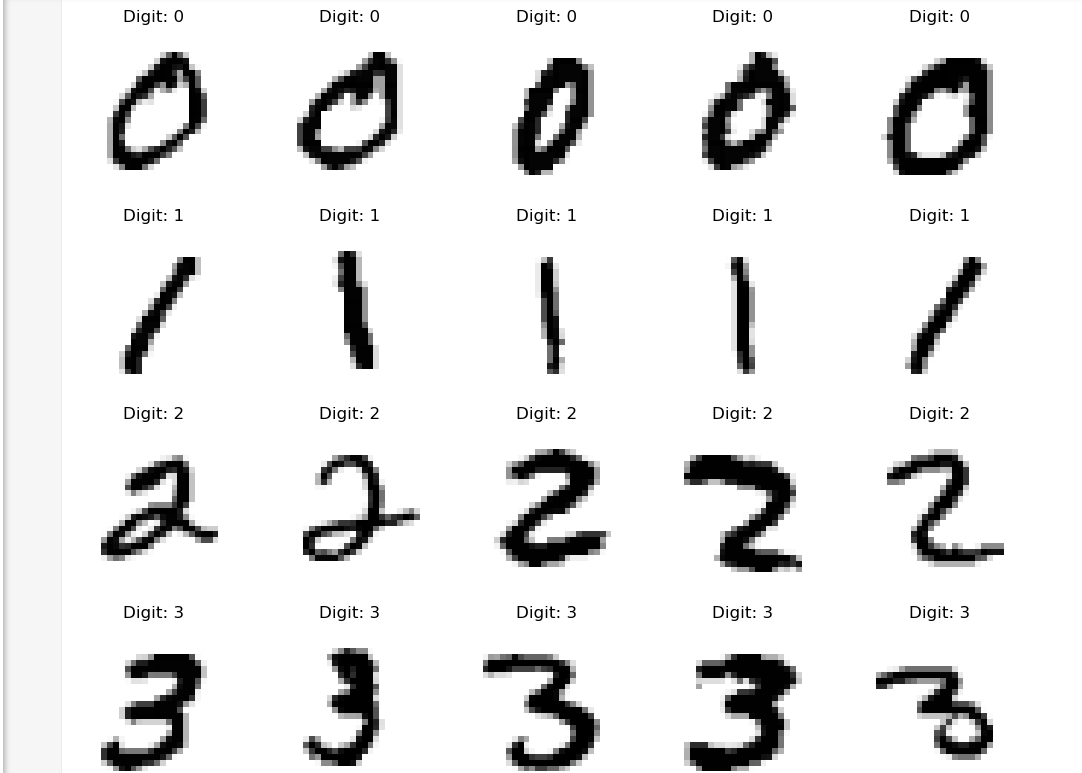
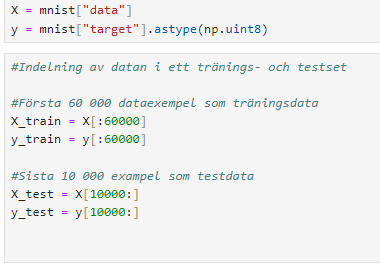
För att utforska datasetet transformerades X-variabelns träningsset till en tvådimensionell variabel, för att sedan kunna visualisera de bilder som datan består av, se figur 3.

MNIST är ett stort dataset, vilket innebär att det tar lång tid för SVM-modellen att träna på datan. Av denna anledning användes enbart de första 10000 instanserna ur träningsdatan till att välja modell. När en modell väl valts användes hela träningsdatasetet för att träna modellen.

För att hitta optimala hyperparametrar användes GridSearchCV för samtliga tre modeller, varefter modellernas förmåga att göra korrekta predikteringar utvärderades. För att göra detta användes accuracy\_score()-funktionen från Scikit learn använts. Funktionen returnerar andelen korrekt predikterade instanser.

Accuracy score användes som det mått som modellerna utvärderades efter. Modellen med högst accuracy score valdes därför för att träna på hela träningssetet med de optimala hyperparametrarna, varefter ett nytt accuracy score räknades ut.

Slutligen användes testdatan på modellen för att utvärdera dess förmåga att korrekt prediktera handskrivna siffror. Även detta mättes med hjälp av accuracy score, men visualiserades även med ett confusion matrix.



Figur 2. Indelning av datan i tränings- och testset

Figur 3. Utdrag av bilder i träningsdatan

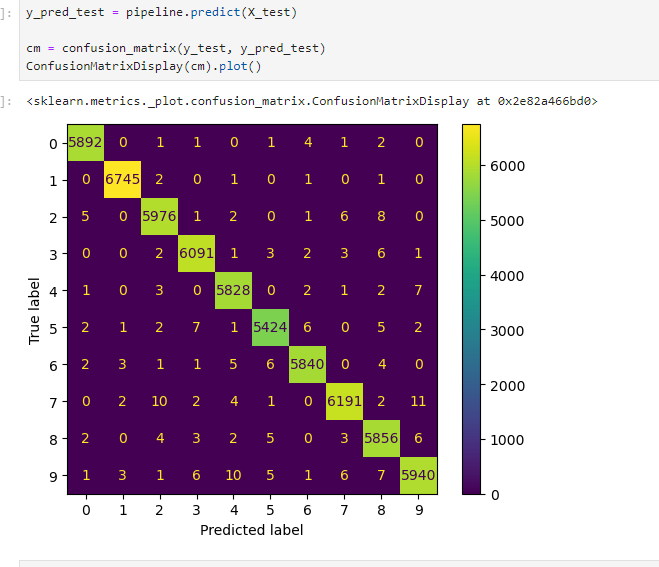
# Resultat och Diskussion

Bland de tre valda modellerna KNN, logistisk regression och SVM var SVM den modell som hade flest andel korrekta predikteringar, se tabell 1. Det skilde dock endast 0,6 procentenheter mellan accuracy score för SVM och logistisk regression. Skillnaden mellan accuracy hos SVM och KNN var däremot större. Där skilde poängen sig åt med 5,65 procentenheter.

Med de valda hyperparametrarna fick vi ett accuracy score på 1,0 vilket innbär att 100 % av det predikterade värdena stämde med SVM-modellen. Trots goda resultat kan detta vara en indikator på överanpassning och att modellen inte är lika framgångsrik på att generalisera på ny data. Testdatan har varit användbar för att utforska detta.

Tabell 2 visar att modellen predikterade korrekta siffror 99,6 % av gångerna på testdatan, vilket är en indikation på att vi inte överanpassat modellen och att den kan göra goda predikteringar.

I vårt confusion matrix kan vi se fördelningen av de felaktiga predikteringar som modellen gjort. Vi kan där utläsa att modellen exempelvis predikterat bild på siffran 9 som en 7 elva gånger, Se figur 4.



Figur 4. Confusion matrix

|  |  |
| --- | --- |
| **Accuracy score på träningsdatan** | |
| KNN | 0,9435 |
| Logistisk Regression | 0,9936 |
| SVM | 1,0 |

Tabell 1: Accuracy score för de valda modellerna på träningsdatan.

|  |  |
| --- | --- |
| **Accuracy score på testdatan** | |
| SVM | 0.9963833333333333 |

Tabell 2. Accuracy score för SVM på testdatan

# Slutsatser

Då SVM genererade det högsta accuracy score, är det den klassificeringsmodell, bland de valda modellerna, med bäst förmåga att prediktera handskrivna siffror korrekt i MNIST-datasetet.

Modellen lyckades prediktera samtliga bilder i träningsdatan korrekt, och 99,6 % av bilderna i testdatan predikterades korrekt.

Bland de visualiseringar av datan vi gjorde under den utforskande delen av projektet, går de att se att handstilarna skiljer sig åt bland de olika bildexemplaren. Är modellen känslig för kurvor och ”brus” kan det innebära att den kan skapa felaktiga predikteringar. Det är dock värt att notera att somliga handstilar kan vara svåra att utläsa även för det mänskliga ögat. För att vidare testa modellens prestanda bör modellen testas på nya exempel av handskrivna siffror.

Modelleringen av MNIST-datan ger en inblick i hur maskininlärning kan användas till klassificeringsproblem, men även bildigenkänning som tillämpningsområde.

# Teoretiska frågor

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

Träningsdatan används för att träna den eller de modeller vi vill utvärdera. Det är träningsdatan som modellen lär sig av för att kunna göra de predikteringar vi är ute efter. Valideringsdatan används under träningsstadiet för att utvärdera hur väl vår modell presterar, och är användbar för att göra justeringar i modellens hyperparametrar för att skapa ännu bättre träffsäkerhet. Testdatan används i det sista steget när vi är färdiga med att träna vår modell. Då testar vi slutligen hur väl modellen presterar. En vanlig indelning av träning-, validering- och testdata är 80 %, 10 % och 10%.

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”?

Hon kan använda sig av K-fold cross validation i sckikit learn på sin träningsdata. Då delas datan upp i k-antal lika delar (folds), tränas på k-1 delar och utvärderas på den återstående delen, som då agerar som valideringsset. Detta upprepas k-gånger där varje ”fold” utgör valideringsset en gång. Sedan räknas ett genomsnittligt mått på prestandan ut. Detta mått väljer vi själva, ex RMSE.

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

potentiella tillämpningsområden?

Regressionsproblem är problem där den beroende variabeln (y) antar kontinuerliga värden som till exempel inkomst, pris eller vikt. Exempel på modeller som kan användas till regressionsproblem är linjär regression, ridge regression och support vector machines.

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

RMSE står för root mean square error och är en formel och ett mått som kan användas för att utvärdera hur säker en maskinlärningsmodell är. I formeln beräknas medelvärdet av skillnaden mellan varje predikterat värde och det sanna värdet i kvadrat, vilket vi slutligen tar roten ur för att kunna tolka resultatet. RMSE beskriver alltså det genomsnittliga ”felet” mellan det sanna och predikterade värdet och används för att utvärdera hur väl en modell presterar.

5. Vad är ”klassificieringsproblem? Kan du ge några exempel på modeller som används och

potentiella tillämpningsområden? Vad är en ”Confusion Matrix”?

Klassificeringsproblem är problem där den beroende variabeln (y) kan delas in i klasser. Några exempel på tillämpningsområden är:

Är epostmeddelandet skräppost eller inte skräppost?

Är siffran på bilden en 0, 1, 2, 3, 4, 5… eller 9?

Är blomman en maskros eller en orkidé?

Exempel på modeller som kan användas till klassificeringsproblem är logistisk regression, K Nearest Neighbors och Support Vector Machines.

En confusion matrix kan användas för att utvärdera klassificeringsmodeller. Det är en tabell med två dimensioner, den sanna och den predikterade. Tabellen visar en fördelning över hur modellen har predikterat och hur det förhåller sig till det sanna värdet.

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

K-means modellen är en typ av icke-väglett lärande som delar in datan i k-antal subgrupper (kluster). Eftersom det är ett icke-väglett lärande är klusterindelningarna på förhand definierade utan antal kluster kan väljas med hjälp av elbow rule. Modellen kan användas till exempelvis kundsegmentering (träningsintresserade, föräldrar, pensionärer, modeintresserade etc)

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

encoding. Se mappen ”l8” på GitHub om du behöver repetition.

Ordinal encoding: Används för variabler som har en inbördes ordning, ex liten, mellan, stor. De kategoriska variabler transformeras till numeriska tillskriva dem en siffra utifrån deras ordning.

Ex: Liten: 0, Mellan: 1, Stor: 2

One-hot encoding: Skapar K antal binära variabler utifrån K kategorier. Beroende på vilken kategori dataexemplen tillhör tilldelas de värdet 0 eller 1 i de olika variablerna. Se nedan exempel på yrkena sotare, polis och snickare. En sotare betecknas med den nya kodningen (1, 0, 0)

|  |  |  |  |
| --- | --- | --- | --- |
| **Yrke** | is\_sotare\_ohe | is\_polis\_ohe | is\_snickare\_ohe |
| Sotare | 1 | 0 | 0 |
| Snickare | 0 | 0 | 1 |

Dummy variable encoding: Liknas vid one hot encoding med använder K-1 dummy variabler vid K kategorier. T ex

|  |  |  |
| --- | --- | --- |
| **Yrke** | is\_sotare\_dummy | is\_polis\_dummy |
| Sotare | 1 | 0 |
| Snickare | 0 | 0 |

I exemplet ovan är snickare varken sotare eller polis och kodas som (0, 0).

8. Göran påstår att datan antingen är ”ordinal” eller ”nominal”. Julia säger att detta måste

tolkas. Hon ger ett exempel med att färger såsom {grön, röd, grön} 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, vi måste alltid tolka datan utefter vårt syfte med vårt arbete. Är syftet att skapa en modell som predikterar priset på bälten i butiker utifrån material, färg och vikt är det kanske rätt väg att gå att se datan som nominal – färgen på bältet har ingen inbördes ordning. Men om syftet handlar om att prediktera vem som vinner i ett slagsmål utifrån färg på karatebälte, vikt och längd så bör datan tolkas som ordinal eftersom i detta sammanhang har vi tillskrivit färgerna en viss ordning beroende på vilken nivå personen uppnått i karate.

9. 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?

Streamlit är en ”open source” miljö där användare enkelt kan skapa webbapplikationer med Pythonscript. Applikationerna kan även göras publika och delas med andra användare.

# Självutvärdering

1. Utmaningar du haft under arbetet samt hur du hanterat dem.

Den främsta utmaningen jag haft under modelleringen av MNIST är tidsåtgången för datorn att träna modellerna. Datorn har krashat flera gången vilket har varit tidskrävande. Detta hanterades dock genom att initialt använda en mindre del av träningsdatan.

1. Vilket betyg du anser att du skall ha och varför.

G då jag utfört alla delar för att uppnå G

1. Något du vill lyfta fram till Antonio?

# Källförteckning

Géron, Aurélien. (2019) *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, Second Edition.* O’Reilly Media, Inc.,

Prgomet, Antonio. (2023) *Support Vector Machines* [Video]. Youtube. <https://www.youtube.com/watch?v=NzGiMYinmog&list=PLgzaMbMPEHEx9Als3F3sKKXexWnyEKH45&index=7>

Prgomet, Antonio. (2023) *Klassificering*. [Video]. Youtube. <https://www.youtube.com/watch?v=-QzFLifZgNw&list=PLgzaMbMPEHEx9Als3F3sKKXexWnyEKH45&index=2>

*Scikit-Learn – Machine Learning in Python* <https://scikit-learn.org/stable/modules/neighbors.html#nearest-neighbors-classification>