Machine Learning

MNIST



Ryan Leyor

EC Utbildning

ML KK2

2025-03

# Abstract

The following report is a machine learning documentation with the objective to train and evaluate models for the classification of handwritten digits from the MNIST dataset.  
Three different models were trained: **Random Forest, Support Vector Machine (SVM), and Stochastic Gradient Descent (SGD)** and their accuracy was compared to determine the best-performing model.  
The Random Forest classifier achieved the highest accuracy of **96.55%** on the test set, outperforming the other models.  
A confusion matrix was analysed to identify common misclassifications, revealing that certain digits, such as **5 and 3 or 9 and 4**, were frequently confused.

# Förkortningar och begrepp

RF : Random forest

SGD: Stochastic Gradient Descent

SVM: Support vector machine

ML: Machine learning

# 

Innehållsförteckning

[Abstract 2](#_Toc193417358)

[1 Förkortningar och begrepp 3](#_Toc193417359)

[2 Inledning 1](#_Toc193417360)

[3 Teori 2](#_Toc193417361)

[3.1 Exempel: Regressionsmodeller 2](#_Toc193417362)

[3.1.1 Exempel: Lasso 2](#_Toc193417363)

[3.1.2 Exempel: Ridge 2](#_Toc193417364)

[3.1.3 Exempel: Elastic Net 2](#_Toc193417365)

[3.2 Exempel: Neurala Nätverk 2](#_Toc193417366)

[4 Metod 3](#_Toc193417367)

[5 Resultat och Diskussion 4](#_Toc193417368)

[6 Slutsatser 5](#_Toc193417369)

[7 Teoretiska frågor 6](#_Toc193417370)

[8 Självutvärdering 7](#_Toc193417371)

[Appendix A 8](#_Toc193417372)

[Källförteckning 9](#_Toc193417373)

# Inledning

Syftet med denna rapport är att genom ett komplett Machinelearning földe från början till slut. Genom att modellera och utvärdera MNIST-data med att träna olika klassifikationsmodeller. Efter utvärdering av olika modeller valdes den bäst presterande modellen och kördes gentemot test-datan.

För att uppfylla syftet så kommer följande frågeställning(ar) att besvaras:

1. Vilken av de valda klassifikationsmodellerna presterar bäst på MNIST-data?
2. Vilka faktorer påverkar modellens noggrannhet och felklassificeringar mest?

# Teori

## MNIST

MNIST databas (*Modified National Institute of Standards and Technology database)* är en storskalig handskriven databas som innehåller digitala nummer. Datasetet består av 60,000 bilder, som är skrivna av gymnasieelever samt författare. Dessa bilder transformerades sedan till en matris på 28x28 pixlar.

MNIST används oftast för upplärning av Machine Learning till ny blivna studenter.

## Klassificeringsmodeller

Eftersom jag ska modellera MNIST databas som uppgift använder jag mig av klassificeringsmodeller och inte regressionsmodeller. Detta på grund av att modellen ska förutsäga en klass (siffrorna 0-9) och inte ett kontinuerligt siffra. Därav har jag utgått från modeller som är anpassade för det.

Dem 3 modeller jag har valt används för att lösa klassifikationsproblem.

### Random Forest Classifier (*RF)*

Randomforest, en ensemblemodell som är baserad på flera slumpliga beslutsträd som tränas tillsammans. Som bilden (*Bild RF*) visar så skapas flera olika tränings grupper på delar av datan för att sen användas för att träna ett eget beslutsträd. Sen när modellen ska prediktera svaret på classfikationen så röstar samtliga predictors och de som får flest röster ”vinner”. (källa:Antonio, wikipedia och Hands on ML bok Kap.7)

A diagram of a training set

AI-generated content may be incorrect. *Bild RF: En beskrivning på hur Random Forest fungerar och tänker.*

*Källa:(s.193) i andra upplagan av Hands on ML bok*

### Stokastisk Gradientnedstigning (*SGD Classfier*)

Detta är en linjär model som presterar egentligen ganska dåligt på dataset som MNIST. Denna classifier är bra för stora dataset eftersom den uppdaterar modellens paramterar genom att utföra ”Minibatch-lärande” där den tar små steg för varje gång den beräknar fel. (Källa: Hands on ML Kap.4 S.124)

### Support Vector Classifier (*SVC)*

Detta är en icka-linjär modell som används för klassificiering med hjälp av support vector machines (SVM). Eftersom jag har använt det för att modellera MNIST-dataset så valde jag att använda ”rbf” kärnan för att få det till icke-linjär, eftersom SVC kan också användas som linjär modell.   
Här maximeras mellanrummet mellan klasserna genom att skapa en icke-linjär gräns. (Källa: Hands on ML s. 157)

# Metod

Arbetet började med att ladda in MNIST-datasetet från openML och skapa ett fullständigt ML-Flöde. Datasetet som består av handskrivna siffror från 0 till 9.

För att förbereda datan, delade jag upp det i X och y variabel.   
Där X: innehåller bilddata representerad som numeriska matriser   
Och y: Motsvarar de faktiska siffrorna i datasetet / det vi vill egentligen predicta.   
  
Sen påbörjade jag träning och validerings processen. För att säkerställa tillräckligt med träningsdata och en robust validering, delade jag datasetet upp enligt följande: 70% träning, 15% validering och 15% test.   
Detta gjorde jag eftersom det är en uppdelning i maskininlärning som säkerställer tillräckligt med data för träning används och en valideringsuppsättning som undviker *overfitting*.   
  
Därefter valde jag att gå vidare med att skapa och träna modellerna jag har valt. Jag valde Random forest (RF) och SGD classifiers på grund av att:   
RF hanterar högdimensionell data utan att behöva skalning  
och *SGD* eftersom den är effektiv och snabb på stora dataset, däremot är den en linjär modell vilket jag hade förväntningar på att den skulle prestera sämre.

Båda modeller tränades på trängsdatan, och deras accuracy score mättes på valideringsdatan.

Eftersom resultatet blev att RF presterade mycket bättre än *SGD,* valde jag att inkludera yttligare en modell (***SVM, som jag implementerade som SVC för att använda en icke-linjär SVM***) som kräver skalning, för att undersöka hur stor skillnaden i prestanda kunde bli. Där visade sig även då att *RF* presterade bättre.

För att yttligare analysera RF-modellens prestanda valde jag att visulisera/skapa en confusion matrix för att identifiera vart den presterar bra och mindre bra. För att se ifall något verkar orimligt. Vilket bedömdes som rimligt då felen den prediktar var förutsebara.   
  
Slutligen testade jag RF-modellen på testadatan, vilket bekräftade att modellens prestanda förblev stabil även på testdata. Därav tog jag bedömningen att modellen är pålitlig och kan användas.

# Resultat och Diskussion

|  |  |
| --- | --- |
| **Modell prestanda** | |
| Random forest Classifier | 0.967 |
| SGD Classifier | 0.865 |
| SVC | 0.961 |

Tabell 1: Accuracy för de utvärderade klassificeringsmodeller (omvandlas till 3 decimaler)

Som vi har nämnt tidigare och nu redovisat i (*Tabell 1),* resultaten visar att *Random Forest* överträffade både SGD och SVC vad gäller nogrannhet med en valideringsnogrannhet på 96.7%. Även om SVC presterade relativt bra, var Random Forest både enklare att använda (ingen skalning krävdes) och hade något bättre resultat.

Resultat skillnaden mellan modellerna kan enkelt förklaras på grund av deras olikheter.

SGD presterade sämre, vilket stämmer överens med mina förväntingar då modellen är linjär och MNIST-data inte är linjärt separerbar.

SVC däremot, som är icke-linjär, lyckades få hög noggrannhet, men krävde längre träningstid och skalning. Något Random Forest kunde göra utan förbehandling, vilket gör den både mer effektiv och träffsäker.

Efter att Random Forest tränades på träning och valideringsdatan så utvärderade jag modellen på testdatan.   
Resultatet stannade stabilt och prestarede med testnogrannhet på 96.55%, vilket tyder på att modelle inte överanpassats (*overfitting*).

Slutligen visualiserade jag modellen med en confusion matrix (*bild 1*), vilket jag valde att göra för att få bättre insikt på vilka siffror som oftast förväxlades. Där jag då bedömde att resultatet verkar rimligt vilket förstärte slutsatsen att jag kan använda modellen.

A graph of a forest confusion matrix

AI-generated content may be incorrect.

Bil 1: Confusion matrix över RF modell

# Slutsatser

Syftet med arbetet har varit att modellera MNIST-datasetet med ett komplett ML-flöde. Därav har jag undersökt mina valda klassifikationsmodeller och kontrollerat vilka som presterat bäst på MNIST-datasetet. Resultatet blev att Random Forest Classifier var mest effektiv, både på noggrannhet och enkelhet då den inte krävde någon förbehandling.

Även om SVC presterade nästan lika bra, krävde den mer skalning och mycket längre träningstid. SGD presterade mindre bra, vilket kan förklaras av att MNIST-datan inte är linjärt separerbar.

Valet av dessa modeller var inte utav ett slump, utan deras olikheter gav mig orsaker.

* *Random Forest* valdes eftersom den fungerar bra med den typen av data där varje datapunkt har jätte många egenskaper och den kräver inte massa justering av hyperparamatrar.
* *SGD* togs med för att testa en linjär modell. Jag ville säkerställa att den faktiskt presterar sämre. Eftersom om den hade visat onormalt hög noggrannhet hade det varit en varningssignal om att något i min kod/träning kunde vara fel.
* *SVC* valdes eftre att jag bekräftade att SGD inte presterar bra, för att få en icke-linjär modell i jämförelsen och 2 olika modeller var en av kraven i arbetet. Eftersom jag visste att MNIST datan inte är linjärt separerbar så var SVC ett logiskt val.

Genom att jämföra de tre modeller, med deras olikheter, kunde jag få en bredare förståelse för både datan och modellernas beteende, vilket har gjort så att mina slutser blivit enklare.

# Teoretiska frågor

Besvara nedanstående teoretiska frågor koncist.

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

Träning är det man använder för att lära modellen, alltså där den får öva och bli bättre. Validering använder man sen för att kolla hur bra modellen funkar på data den inte har tränat på, så man kan testa och jämföra olika modeller för att sen välja den bästa.  
Sen går man vidare med att använda modellen för att testa den på testadatan för att se hur bra den verkligen är.

1. 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 ”validerings-dataset”?

Julia kan göra något som kallas för cross validation på hennes modeller. Det skulle innebära att hon delar upp träningsdatan i flera delar för att träna den på hennes valideringsdata och resterande data.. Detta gör hon flera gånger för att få flera siffror som hon kan ta medelvärde. Den modell som får bäst genomsintt är då ett bra val att fortsätta med.

1. Vad är ”regressionsproblem? Kan du ge några exempel på modeller som används och potentiella tillämpningsområden?  
   När man har mål på att förutsäga nummer, priser, tempratur och variabler som är kontinuerliga osv.. så handlar det som regressionsproblem.

* Exempel på Modeller: linjär, ridge och lasso – regression.

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

RMSE används för att hitta en modells genomsnitt fel. Den används för att se hur bra egentligen en regressionmodell fungerar. Det man vill ha är lägre RMSE.  
Vill man göra bättre val av modeller kan man använda RMSE för att se vilken modell har lägst RMSE.

1. Vad är ”klassificieringsproblem? Kan du ge några exempel på modeller som används och potentiella tillämpningsområden? Vad är en ”Confusion Matrix”?  
   När man har som mål att förutsäga kategoriska beslut som ja/nej, poitiv/negativ.   
   - Exempel på Modeller: Random Forest Classifier och Support Vector Classifier

Confusion Matrix är som en tabell som hjälper en att se hur en modell har gissat. Som i MNIST-dataset så kan man se hur många gånger den har gissat på nummer 1 som nummer 1 och hur många gånger den har gissat på att nummer 1 är nummer 9 (hur många rätt och hur många fel)

1. Vad är K-means modellen för något? Ge ett exempel på vad det kan tillämpas på.  
   K-means används för klustring av data. Det är en ”*unsupervised”*  modell som används för att hitta mönster i data genom att del in den i kluster/k grupper. Sen tar modellen och väljer ”centroids” (ett centrum för varje grupp) och placerar varje datapunkt i det kluster vars centrum ligger närmast. Sen upprepas det tills klustren är stabila.

Ett exempel skulle vara att man äger en butik där man har en viss data om sina kunder (genom medlemskap osv) , som visar hur mycket vissa kunder handlar och hur ofta med K-means kan man dela upp kunderna som ”handlar ofta” och ”handlar sällan” utan att dela upp någon i specifik grupp så ta k-means och kollar på kundens beteende och bedömer själv vilken grupp den tillhör

1. Förklara (gärna med ett exempel): Ordinal encoding, one-hot encoding, dummy variable encoding. Se mappen ”l8” på GitHub om du behöver repetition.  
   När man har ett sett av data som inte har en numersik ordning, som Ex: Färger, storlek eller biltillverkare(Ex med data som har ordning: alfabetet och årtal), så måste man ge den datan särskilda nummer för att modeller ska kunna jobba med och för att göra detta kan man använda sig av dessa metoder:

Ordinal encoding: säg att vi har värden liten, mellan och stor. Då enligt ordinal encoding så delar vi upp desa som Liten = 0 , Mellan = 1 och stor = 2.. Här spelar ordningen roll.

one-hot encoding: Om vi har data som 3 färger Orange, röd och blå. Då skapas 3 kolumner och en av dem får numret 1 om det stämmer in på kategorin.. låt säga Röd = [1,0,0], blå = [0,1,0] och orange = [0,0,1]

dummy variable encoding: Gör nästan samma sak om one-hot encoding men lämnar en kategori. Det gör för att tillexempel kunna utföra linjär algebra och man undviker överflödig information.

Exempel på hur det då ser ut: Röd[1,0], Blå[0,1] och orange [0,0]

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 i och med att färger inte har någon inbördes ordning/nominal. Att säga att färgen röd vackrast skulle innebära att man skapar en ordning då den färgen blir helt plötsligt bästa och får högt värde.. då blir datan ordinal

1. Kolla följande video om Streamlit: <https://www.youtube.com/watch?v=ggDa-RzPP7A&list=PLgzaMbMPEHEx9Als3F3sKKXexWnyEKH45&index=12>  
   Och besvara följande fråga:  
   Vad är Streamlit för något och vad kan det användas till?

Streamlit är en python bibliotek som man använder för att skapa webbapplikation som man sen publicerar publikt. I det här fallet skulle man använda streamlit för att skapa en app på webben som prediktar vilka siffror man har skrivit för hand

# Självutvärdering

1. Utmaningar du haft under arbetet samt hur du hanterat dem.   
   Att hitta korrekta ord för att förklara vad man har gjort och hur man har tänkt.. Svåraste biten har varit att försöka att inte repetera saker man redan har skrivit om och om igen.. Jag försökte ge mig in på APP skapandet och tror att min rapport hade sätt mycket bättre ut om jag hade lyckats för att jag hade haft mera att skriva och gå in djupare på.
2. Vilket betyg du anser att du skall ha och varför.

G- Eftersom jag anser mig själv ha lyckats /stasat på dem kriterierna som krävs för att uppnå betyget

1. Något du vill lyfta fram till Antonio?  
   Detta var ett väldigt roligt utbildning och jag uppskattar verkligen lektionen som du håll i under 2025-03-04 då den var väldigt informativ och enkelt förklarad.. Detta förenklade väldigt mycket att få grunderna för och förstå boken bättre och vad hela syftet är. Jag uppskattar sådana detaljerade information.  
   Boken var däremot väldigt svår för en person som jag, då jag har väldigt enkelt för och fastna i detaljer för att jag vill förstå exakt vad jag läser och vad det betyder.. vilket har lett till flera timmar slösade på att förstå koder och deras beteende. En svår bok men uppskattande undervisningar som förenklade..

Detta leder tyvärr till att jag tar längre tid på mig att förstå och gå vidare och därav hamnar efter i planeringen.. därav sena inlämningar som denna.. Jag uppskattar verkligen förståelse kring deta

# Källförteckning

Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow:

<http://14.139.161.31/OddSem-0822-1122/Hands-On_Machine_Learning_with_Scikit-Learn-Keras-and-TensorFlow-2nd-Edition-Aurelien-Geron.pdf>

Scikit Learn - Random Forest Classifier:

<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>

Confusion Matrix:

<https://www.w3schools.com/python/python_ml_confusion_matrix.asp>

Scikit Learn – SVM:

<https://scikit-learn.org/stable/auto_examples/svm/plot_svm_kernels.html#sphx-glr-auto-examples-svm-plot-svm-kernels-py>

<https://www.baeldung.com/cs/svm-feature-scaling>

MNIST-dataset

<https://en.wikipedia.org/wiki/MNIST_database>