Kunskapskontroll 2

Maskininlärning



Anita Kongpachith

EC Utbildning

Rapport - Maskinlärning

2024-03-15

# Abstract

This paper examines the MNIST database of handwritten digits used for training image processing. By using machine learning methods, it gets a deeper understanding of what AI technology is really about. Being creative and create different models to predict if the accuracy score is at least 80 % is one of the questions to be answered. The results gets a pleasant astonishment, but also learning how the models are performing can be visualise by a confusion matrix. Machine learning are powerful tools for solving problems and that is why this database is about to explore by this algorithms.

Innehåll

[Abstract 2](#_Toc162898753)

[Förkortningar och Begrepp 3](#_Toc162898754)

[1 Inledning 1](#_Toc162898755)

[1.1 Syfte 1](#_Toc162898756)

[2 Teori 2](#_Toc162898757)

[2.1 Modeller 2](#_Toc162898758)

[2.1.1 Random Forest Classifier 2](#_Toc162898759)

[2.1.2 SVM / LinearSVC 2](#_Toc162898760)

[2.1.3 K – nearest neighbors (KNN) 2](#_Toc162898761)

[2.2 Confusion matrix 2](#_Toc162898762)

[2.3 Voting Classifier 2](#_Toc162898763)

[3 Metod 3](#_Toc162898764)

[3.1 Databas 3](#_Toc162898765)

[3.2 Explanatory Data Analysis (EDA) 3](#_Toc162898766)

[3.2.1 Tränings set / Validation set 3](#_Toc162898767)

[3.2.2 Voting Classifier 3](#_Toc162898768)

[3.3 Vald modell 4](#_Toc162898769)

[4 Resultat och Diskussion 4](#_Toc162898770)

[5 Slutsatser 7](#_Toc162898771)

[6 Teoretiska frågor 8](#_Toc162898772)

[7 Självutvärdering 10](#_Toc162898773)

[Appendix A 12](#_Toc162898774)

[Källförteckning 13](#_Toc162898775)

# Inledning

Om man kollar för 20 – 30 år sedan så har digitaliseringen förändrats otroligt mycket, då teknologin har utvecklats och har idag åstadkommit med något som heter AI som står för artificiell intelligens. Med hjälp av AI som har så kan datorer tolka och förstå omvärlden bättre, exempel kan vara att datorn ska kunna resonera, planera och kunna lära sig av data. Ett annat ord är att få datorer var lite mer lik människor. För att detta ska fungera så byggs det genom en metod som kallas för maskininlärning som är en delmängd av AI, då algoritmer tränas för att hitta mönster och korrelationer i en stor mängd data och sedan fatta de bästa besluten och förutsägelserna utifrån den analysen. Så i helhet i stället för att skriva direkta program med instruktioner så skapar vi program som gör att datorer lär sig utan att vara explicit programmerade.

Genom maskininlärning så ska en databas som heter MNIST undersökas och modelleras i detta arbete. MNIST baseras på handskrivna siffror som tillämpas genom att träna olika bildbehandlingar.

## Syfte

Syftet med denna rapport är att skapa en maskininlärningsmodell som tillämpas av MNIST databas för att prediktera en given bild av en handskriven siffra i en av tio klasser som representerar heltalsvärden 0 till 9.

Med fortsättning att skapa koder för att prediktera en handskriven siffra som tas ifrån en mobilkamera eller med en webbkamera som utförs genom Streamlit applikationen.

För att uppfylla syftet så kommer följande frågeställningar att besvaras:

1. Att skapa en modell för att kunna prediktera MNIST med minst 80% noggrannhet?
2. Kan modellen sedan kunna prediktera bilder från en mobilkamera eller med en webbkamera?

# Teori

## Modeller

### Random Forest Classifier

Random Forest Classifier är en supervised learning algoritm, som är en ensemble av beslutsträd som oftast tränas genom bagging (slumpmässigt urval med återläggning). Random forest använder medelvärde för att förbättra den prediktiva noggrannheten (accuracy) och kontrollera noggrannheten. (Géron, 2019) (Scikit-learn)

### SVM / LinearSVC

SVM som står Support Vector Machine, vilket kan hantera modeller som har klassificeringsproblem och regressionsproblem. SVM använder data punkter (vektorer) som är refererat som support vectors och har i sin huvudsak att hitta den ideala hyperplanet som har störst margin som kan skapa diskreta klasser som delas upp till en n-dimensionell rum. (Géron, 2019) (Scikit-learn)

LinearSVC står för Linear support vector classification, som försöker hitta hyperplanet för att maximera avståndet mellan de klassificerade urval. Och kan hantera större urval bättre på grund av det finns fler alternativ förlust funktioner och påföljder.

### K – nearest neighbors (KNN)

Är en icke parametrisk algoritm som använder närhet för att göra klassificeringar och prediktioner om gruppering av en enskild datapunkt, det vill säga antagande av liknade datapunkter som kan hittas nära varandra. Och kan tillämpas för både regressionsproblem och klassificeringsproblem. (Géron, 2019) (Scikit-learn)

## Confusion matrix

Confusion matrix är en matris som visar antalet korrekta och felaktiga instanser baserat på modellens prediktioner.

## Classification Report

Classification report är en text rapport som visar olika klassificerings metrics som precision, recall, F1 och support scores.

Precision – Andelen av de positiva prediktionerna som faktiskt ät korrekta

Recall – Andelen av den positiva klassen som predikteras korrekt

F1 score – agerar som ett harmoniskt medelvärde för precision och recall, där optimalast poängen är 1,0 och sämsta poängen är 0,0

Support – är antalet faktiska förekomster av klassen i den givna datamängden

## Voting Classifier

Voting Classifier är en metod som tränar många ensemble av olika modeller för att sedan prediktera ett resultat baserat på högsta majoriteten av röster. Tanken bakom är att skapa en enda modell som tränar efter dessa modeller och förutsäger resultatet som utgår från deras kombinerade majoritet av rösterna för varje output klass. (Géron, 2019) (Scikit-learn)

# Metod

## 

## Databas

I detta arbete användes databasen MNIST som innehåller bilder med handskrivna siffror med 784 features. Det kan splittras i tränings set med 60,000 exemplen och test set med 10,000 exemplen.

MNIST är en delmängd som kommer från en större databas som heter NIST. Siffrorna har blivit storleksnormaliserad och centrerad i en bild med fast storlek. Originalet var svartvita bilderna från NIST normaliserades så att det skulle passa in i en 28x28 pixlar och anti\_aliasing, vilket resulterade att gråskalenivåer introducerades.

Detta laddades ner genom Scikit\_learns ramverk ”fetch\_openml” för att skapa modeller och testning.

## Explanatory Data Analysis (EDA)

Datasetet splittrades innan modeller skapades för att sedan testas på tränings setet. Funktionen StandardScaler tillämpades innan tränings set för att standardisera funktioner genom att ta bort medelvärden och skala till enhetsvariansen. Genom StandardScaler så kan modeller få en förbättring av prestandan.

### Tränings set / Validation set

* Olika modeller skapades och tränades på skalad tränings data.
* Sedan predikterades de skalade modellerna på validations data.
* Noggrannheten (accuracy score) beräknades efter prediktionen för att se vilken modell som presterade bäst.
* Visualisering av prestandan på modellen illustrerades genom en confusion matrix för att få en uppfattning.

### Voting Classifier

För att vara säker på att vilka modeller som presterade bäst så skapades voting classifier som tränades på skalad tränings data och sedan evaluerades på validations data genom en score metod för resultat.

## Vald modell

Utgående från olika metoder så valdes random forest modellen för att evaluera på skalad test data, då modellen hade det mest optimala resultatet jämfört med de andra modellerna. Random forest predikterades på skalad test data och tillämpades genom accuarcy score för resultat. Sedan visualiserades modellen genom en confusion matrix för att se en tolkning av prestandan.

# Resultat och Diskussion

Resultatet av modellerna presterade väldigt bra med små skillnads marginal, men i jämförelse av samtliga resultat kan man se att random forest hade högst noggrannhet (accuracy). Även följande resultat visualiserades genom en confusion matrix och classification report för varje modell. Tolkningen var inte så stor skillnad på noggrannheten och de andra metrics, men ser att random forest hade det optimala resultatet.

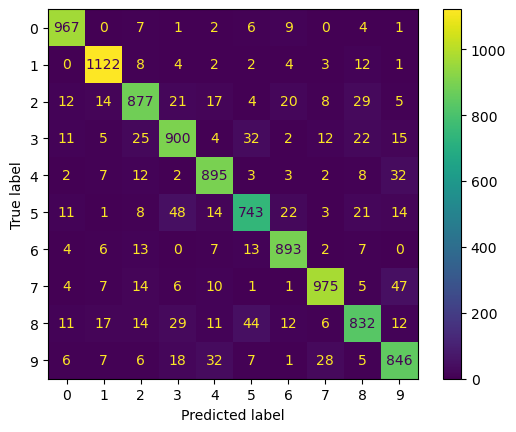
|  |  |
| --- | --- |
| **Modeller** | **Accuracy score** |
| Random forest | 0,969 |
| LinearSVC | 0,905 |
| KNN | 0,943 |

Confusion matrix för Random Forest

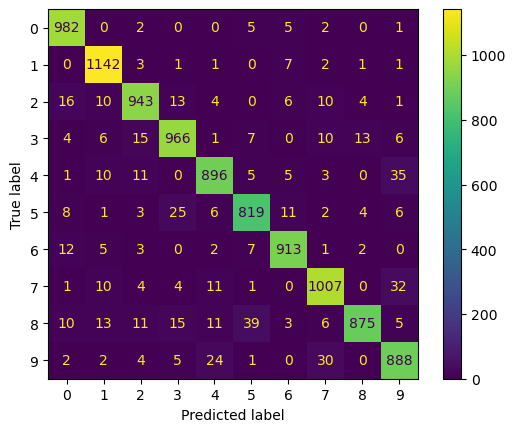
En bild som visar text, skärmbild, kvadrat, diagram

Automatiskt genererad beskrivning

Confusion matrix för Linear SVC



Confusion matrix för KNN



För mer träffsäkerhet att random forest presterade optimalast så tillämpades en Voting Classifier för att se om det presterade bäst till skillnad från de andra modellerna. Av resultatet så ser man att det random forest hade högst majoritet av rösterna, vilket gör att den valda modellen som kommer att evalueras på test data är random forest.

|  |  |  |
| --- | --- | --- |
| Random forest | LinearSVC | KNN |
| 0,9692 | 0,9074 | 0,9431 |

Resultatet av Random Forest modellen på test data fick en accuarcy score på 0,964 och en confusion matrix har visualiserats för en bättre uppfattning. Skillnaden från validations data är marginellt då resultatet var 0,969 och även confusion matrix var liknande.

En bild som visar skärmbild, kvadrat, mönster, Färggrann

Automatiskt genererad beskrivning

# Slutsatser

Går det skapa en modell för att kunna prediktera MNIST med minst 80% noggrannhet?

Av resultatet som har tagits fram så är svaret ja. Över förväntan så presterade modellerna som skapades väldigt bra på minst 90 % noggrannhet, och av samtliga modeller så hade random forest högst prestation med 97% noggrannhet vilket är ett bra resultat för att prediktera en bild med handskriven siffra.

Genom MNIST databas har man fått en mer grundlig förståelse för maskininlärning att programmera så data lär sig att utföra uppgiften.

Det finns många olika sätt att skapa koder och metoder som ger samma resultat, vilket kan vara förvirrande när man ska börja skapa modeller och vilka tillämpningar som ska användas. Vissa metoder tar längre tid innan ett resultat visas vilket gör att programmeringen tar lång tid. Av hela processen kan detta tänkas vara större utmaningar med folk som arbetar med maskininlärning för att data ska lära sig och fatta beslut vilket är komplex och kan vara svårt att förstå. En reflektion som kom upp när detta arbete skrevs.

Kan modellen sedan kunna prediktera bilder från en mobilkamera eller med en webbkamera?

Ja det kan man, men det beror också på vilken vinkel och hur bra kameran tar upp ljuset. I detta fall har ingen lösning kommit på än då kameran inte predikterade siffrorna rätt, problemet kommer att åtgärdas tills det är löst och sifforna kan prediktera siffrorna rätt.

Av ingen erfarenhet tog detta tid att förstå och har därför inte har fungerat än men med tiden så ska det fungera och man kan vidareutveckla om man till exempel i stället för att ta en bild så kan man testa att skriva siffra förhand med dator musen och se om det går att prediktera siffran.

# Teoretiska frågor

1. Tränings data set – Används till att för skapa och träna olika modeller.

Validerings data set – Används till för att utvärdera de olika modellerna för att sedan välja ut den optimala modellen.

Testdata set– Används till för att testa den valda modellen för att få en uppskattning om data. Innan dess så tränas modellen om igen på träningsdata och valideringsdata för att sedan testas.

1. Julia kan använda cross validation för att ta fram vilken av de tre modellerna som är mest optimala. Genom cross validation som använder olika modeller som delas upp i tränings set och test set, som sedan genererar poängsättningar så kan man utvärdera och jämföra vilken modell som presterar bäst.
2. Regressionsproblem är beroende variabler som har kontinuerliga värden, det vill säga tillämpar numeriska problem. Ett exempel på regressionsproblem kan vara varuhandling, priset på produkter. Resultatet är slutsumman som beror på antal produkter och priset. Några modeller som använder regressionsproblem är exempelvis linjär regression, support vector machines (SVM) och beslutsträd.
3. RMSE tolkar det genomsnittliga skillnaden som förväntas mellan de förutsagda värden och de faktiska värden, det vill säga tillämpas för att beräkna prediktioners medelavstånd till dem sanna värdena.
4. Klassificeringsproblem koncentrerar sig på variabler som har diskreta värden, det vill säga består av kategorier och icke numeriska problem. Ett exempel på klassificeringsproblem kan vara binära klassifikation, är det ett ”falskt konto” eller ”äkta konto” på Twitter. Några modeller som tillämpar klassificeringsproblem är exempelvis logistisk regression, random forest och support vector machines (SVM).

Confusion matrix är en matris som visar antalet korrekta och felaktiga instanser baserat på modellens prediktioner. Exempelvis kan vara en bild på en hund (ja/nej).

En confusion matrix innehåller:

* True Positives (TN) – inträffar när modellen predikterar positivt och det är korrekt.
* True Negatives (TN) – inträffar när modellen predikterar negativt och det är korrekt.
* False Positives (FP) – inträffar när modellen predikterar positivt och det är falskt.
* False Negatives (FN) – inträffar när modellen predikterar negativt och det är falskt.

1. K-means används inom unsupervised learning, då det är en algoritm som klassificerar data punkter till gruppering (kluster) av den information man har om data. Det vill säga k-means försöker hitta varje klusters centroid observationer (centrum i klustret) för att sedan klassificera alla data punkter till de kluster som har närmast avstånd till centrum. Ett exempel som kan tillämpas är dokumentation klustring, då man har multipla dokument och behöver gruppera liknade dokument tillsammans. Klustring hjälper till att sortera dessa dokument då de som har liknande dokument är i samma kluster.
2. Ordinal encoding - är en kategorisk data som är en rankad ordning som har ett numeriskt värde och har en relation till varandra. Ett exempel kan vara olika stadier i grundskolan, då lågstadiet = 1, mellanstadiet = 2 och högstadiet = 3.

One-hot encoding – omvandlar kategorier till vektorer som innehåller antingen 0 eller 1. Det skapar många kolumner vilket gör att processen blir långsammare, då numeriska värden blir för stort och har ingen relation till varandra.

Ett exempel på en data:

|  |  |
| --- | --- |
| ID | Färg |
| 1 | Röd |
| 2 | Grön |
| 3 | Blå |
| 4 | Röd |

Efter applicering av one-hot encoding:

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Röd | Grön | Blå |
| 1 | 1 | 0 | 0 |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 0 | 1 |
| 4 | 1 | 0 | 0 |

Dummy variable – Är binära variabler som visar kvalitativ information i en regressionsanalys. Det vill säga som enkelt förklarar huruvida de individuella observationerna tillhör till en särskild kategori. Ett exempel kan vara att kategorisera kön till man eller kvinna då man = 0 och kvinna = 1.

1. Göran har rätt, beroende på data kan man identifiera om det är nominal eller ordinal variabel. Som Julias exempel på färger så är det exempel på nominal variabel, då det är kategoriska variabler och som inte har någon rangordning. Att man har en röd skjorta och är vackrast på festen är inte en ordinal variabel, då färger inte har en relation till varandra utan mer en individuell fråga om man tycker att personen som har en röd skjorta på sig är vackrast eller inte.
2. Streamlit är ett ramverk för appar med öppen källkod, då de används till för att skapa dataapplikationer genom python skript. Vilket gör att data scientist och maskininlärnings ingenjörer kan tillämpa detta för att enkelt skapa webb appar genom python.

# Självutvärdering

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

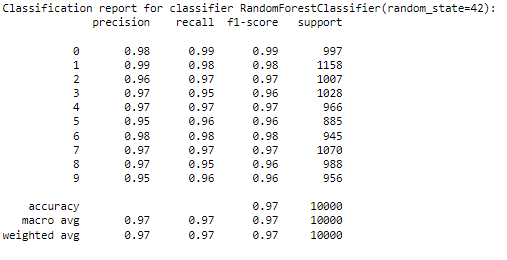
Svårt att komma i gång då det finns många olika modeller att välja på och hur man ska programmera och hur mycket man ska fördjupa sig i det. Streamlit har varit en utmaning som jag inte har lyckats att lösa än och därför inte inkluderat med i arbetet.

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

Jag lyckades inte med streamlit än så godkänd är förhoppningen i alla fall.

1. Något du vill lyfta fram till Antonio?

# Appendix A



En bild som visar text, skärmbild, nummer, Teckensnitt

Automatiskt genererad beskrivning

En bild som visar text, skärmbild, nummer, Teckensnitt

Automatiskt genererad beskrivning

# Källförteckning

Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow (2nd ed.). O’Reilly Media, Inc

<https://www.geeksforgeeks.org/>

<https://scikit-learn.org/stable/index.html>