Maskininlärning

Modellera MNIST-Dataset



Melike Koyuncu

EC Utbildning

Maskininlärning

202403

# Abstract

In this project, I worked with the MNIST dataset to create and evaluate machine learning models. The process compares two machine learning models, a random forest model and a logistic regression model to classify handwritten digits. I also used GridSearchCV() to find the best parameters for the models. By doing these steps, achieved an accuracy of % 93.7 with the random forest model on the test set which was the higher accuracy than logistic regression model. The results show that the random forest model outperforms the logistic regression model for this task.

**Skapas automatiskt i Word genom att gå till Referenser> Innehållsförteckning.**

Innehållsförteckning

[Abstract 2](#_Toc161935858)

[1 Inledning 1](#_Toc161935859)

[1.1 Syfte och frågeställningar 1](#_Toc161935860)

[2 Teori 2](#_Toc161935861)

[2.1 Logistisk Regression 2](#_Toc161935862)

[2.1.1 Välja Hyperparameter för Logistisk Regression 2](#_Toc161935863)

[2.2 Random Forest 3](#_Toc161935864)

[2.2.1 Välja Hyperparameter för Random Forest 3](#_Toc161935865)

[2.3 GridSearchCV 3](#_Toc161935866)

[3 Metod 4](#_Toc161935867)

[4 Slutsatser och Resultat 7](#_Toc161935868)

[5 Teoretiska frågor 8](#_Toc161935869)

[6 Självutvärdering 11](#_Toc161935870)

[Appendix A 12](#_Toc161935871)

[Källförteckning 12](#_Toc161935872)

# Inledning

Maskininlärning är en gren av artificiell intelligens som används många områden för att lösa en mängd olika problem, inklusive bildigenkänning, röstigenkänning, naturlig språkbehandling och prediktiv analys.

## Syfte och frågeställningar

Syftet med denna rapport är att skapa och utvärdera maskininlärningsmodeller för att klassificera handskrivna siffror i MNIST-datasetet. För att uppfylla detta syfte kommer följande frågeställningar att besvaras:

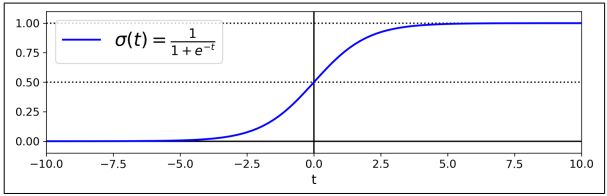
1. Vilken är den bästa maskininlärningsmodellen för att klassificera handskrivna siffror i MNIST-datasetet?
2. Vilka är de optimala hyperparametrarna för den valda modellen?

Genom att besvara dessa frågeställningar kommer att kunna utvärdera olika maskininlärningsmodeller och tekniker för att klassificera handskrivna siffror i MNIST-datasetet och identifiera den bästa modellen med optimala hyperparametrar. Detta kan bidra till att förbättra vår förståelse för hur maskininlärningsmodeller fungerar och hur man optimerar deras prestanda för bildigenkänning.

# Teori

## Logistisk Regression

Logistisk regression används vanligtvis för att uppskatta sannolikheten för att en instans tillhör en viss klass (t.ex. vad är sannolikheten att det här e-postmeddelandet är skräppost?). Om den uppskattade sannolikheten är större än 50 %, förutsäger modellen att instansen tillhör den klassen (kallad positiv klass, märkt "1"), och annars förutsäger den att den inte gör det (dvs. den tillhör den negativa klassen). , märkt "0"). Detta gör den till en binär klassificerare. (Géron, 2019, s. 142)



Figur 1. Logistisk funktion

### Välja Hyperparameter för Logistisk Regression

Hyperparametrar för logistisk regression kan väljas genom en process som kallas hyperparameter tuning. Detta innebär att man väljer ett intervall av värden för varje hyperparameter och sedan tränar och utvärderar modellen med varje kombination av hyperparametrar för att hitta den kombination som ger bäst prestanda på en valideringsuppsättning. Några vanliga hyperparametrar för logistisk regression inkluderar regulariseringsparametern, inlärningshastigheten och antalet iterationer. Regulariseringsparametern styr mängden av regularisering som tillämpas på modellen, vilket hjälper till att förhindra överanpassning. Det finns flera parametrar att välja men in this project används C parametern. Det är viktigt att veta det måste vara ett positivt flyttal. Mindre värden specificerar starkare regularisering och höga värden talar om för modellen att ge hög vikt åt träningsdatan.

## Random Forest

Random Forest är en kraftfull ensemble-metod som kombinerar flera beslutsträd för att förbättra prediktiv noggrannhet och minska överanpassning. För klassificeringsuppgifter är utdata från den slumpmässiga skogen den klass som väljs av de flesta träd. För regressionsuppgifter returneras medelvärdet eller medelvärdet för de individuella träden. Random forest korrigerar beslutsträdens vana att överanpassa sin träningsuppsättning.

### Välja Hyperparameter för Random Forest

Här är två viktiga hyperparametrar som används i projektet för Random Forest:

Första parameter är ’n\_estimators’ som styr antalet träd inuti klassificeraren. Värdet n\_estimators anger hur många träd som ska användas. Ju fler träd, desto mer robust är modellen. Standardvärdet är 100, men det kan justeras. Och den andra parameter är ’max\_depth’ som begränsar djupet på varje beslutsträd. Det styr den maximala höjden till vilken träden inne i skogen kan växa. Att öka trädets djup kan förbättra noggrannheten upp till en viss punkt, men utöver det kan det leda till övermontering.

## GridSearchCV

GridSearchCV är en metod för att söka efter de bästa hyperparametrarna för en maskininlärningsmodell och används i project för båda modellerna. Det innebär att man söker efter de bästa kombinationerna av hyperparametrar genom att träna och utvärdera modellen på olika kombinationer av hyperparametrar. GridSearchCV är en vanlig metod för att optimera modeller eftersom det är en systematisk och effektiv metod för att hitta de bästa hyperparametrarna.

# Metod

I detta projekt, skriven kod i Python för att träna och utvärdera två maskininlärningsmodeller (logistisk regression och random forest modell) på MNIST-datauppsättningen.

Här är en uppdelning av koden:

Delar upp data i tränings-, validerings- och testset och skalar funktionerna med StandardScaler.

Tränar och utvärderar en logistisk regressionsmodell med hjälp av GridSearchCV för att hitta de bästa hyperparametrarna.

Tränar och utvärderar en random forest modell med hjälp av GridSearchCV för att hitta de bästa hyperparametrarna.

Importerar nödvändiga bibliotek och laddar MNIST-datauppsättningen.

En bild som visar skärmbild, diagram, linje, kvadrat

Automatiskt genererad beskrivning

Figur 2. En bild från MNIST-datasetet

Med hjälp av kod hämtas en bild från MNIST-datasetet genom att använda index 7. Sedan omvandlas bilden till en 28x28 storlek och visas med hjälp av matplotlibs imshow-funktion. Funktionen cmap=mpl.cm.binary används för att visa bilden i svartvitt. Således visar en bild av en handskriven siffra från MNIST-datasetet.

En bild som visar skärmbild, text, kvadrat, Färggrann

Automatiskt genererad beskrivning

Figur 3. Confusion Matrix

I koden skrivs formen på träningsdatan X\_train och dess etiketter y\_train ut. Detta görs för att visa antalet exempel och funktioner i träningsdatan och antalet etiketter i y\_train.

Sedan definieras en funktion display\_confusion\_matrix som tar in två argument, y\_test och y\_pred. Funktionen beräknar förvirringsmatrisen för dessa två etikettuppsättningar med hjälp av confusion\_matrix-funktionen från sklearn.metrics. Därefter skapas en ConfusionMatrixDisplay-objekt och dess plot-funktion används för att visa förvirringsmatrisen.

Till sist anropas display\_confusion\_matrix-funktionen med y\_test och y\_pred som argument. Detta visar förvirringsmatrisen för den bäst presterande modellen på testuppsättningen.

Sammantaget används den här koden för att jämföra prestandan för två olika maskininlärningsmodeller på MNIST-datauppsättningen och avgöra vilken som presterar bäst.

För att svara på frågeställningarna användes en metod som kallas för GridSearchCV. Detta är en teknik som används för att hitta de optimala hyperparametrarna för en maskininlärningsmodell genom att systematiskt testa olika kombinationer av hyperparametrar och utvärdera modellers prestanda på en valideringsuppsättning.

För att använda GridSearchCV i detta fall, delade MNIST-datasetet upp i en träningsuppsättning, en valideringsuppsättning och en testuppsättning. Träningsuppsättningen användes för att träna modellerna, valideringsuppsättningen användes för att utvärdera modellerna och hitta de optimala hyperparametrarna, och testuppsättningen användes för att utvärdera modellernas prestanda på oberoende data.

För att svara på den första frågeställningen tränades två olika maskininlärningsmodeller - en logistisk regressionsmodell och en slumpmässig skogmodell - på träningsuppsättningen. Sedan utvärderades modellerna på valideringsuppsättningen för att bestämma vilken modell som presterade bäst.

För att svara på den andra frågeställningen användes GridSearchCV för att hitta de optimala hyperparametrarna för varje modell. Genom att testa olika kombinationer av hyperparametrar, såsom C-värdet för logistisk regression och antalet träd och djupet för slumpmässig skog, kunde de bästa hyperparametrarna för varje modell hittas.

Det material som används för att svara på frågeställningarna var MNIST-datasetet och Python-programmeringsspråket med hjälp av olika bibliotek som scikit-learn, numpy, pandas och matplotlib.

# Slutsatser och Resultat

Här är några slutsatser och resultat som kan lyftas fram genom arbetet:

* Antalet träningsdata påverkar modellernas prestanda. Genom att minska antalet träningsdata kan man få en snabbare träningsprocess men det kan också leda till en sämre prestanda på validerings- och testuppsättningarna.
* Både logistisk regressionsmodellen och random forest modellen presterar bra på MNIST-datasetet, men den random forest modellen verkar prestera något bättre på testuppsättningen, med en noggrannhet på 0,943. Förvirringsmatrisen visar att modellen presterade bra på de flesta siffror, men hade vissa svårigheter att skilja mellan 4 och 9.
* Antalet korsvalideringsveck kan påverka modellernas prestanda. Genom att öka antalet veck kan man få en mer tillförlitlig uppskattning av modellernas prestanda men det kan också leda till en längre träningsprocess.
* Värdet på C-värdet för logistisk regressionsmodellen kan påverka modellens prestanda. Genom att testa olika värden på C-värdet kan man hitta det optimala värdet för modellen.
* Antalet träd och djup för random forest modellen kan påverka modellens prestanda. Genom att testa olika antal träd och djup kan man hitta de optimala hyperparametrarna för modellen.
* Olika kombinationer av hyperparametrar kan påverka modellernas prestanda. Genom att testa olika kombinationer av hyperparametrar kan man hitta de optimala hyperparametrarna för modellerna.
* Arbetet visar också hur man kan använda GridSearchCV för att hitta de bästa hyperparametrarna för varje modell. Genom att använda GridSearchCV kan man optimera modellerna för att uppnå högre noggrannhet.
* Jag vill påminna om vikten av att fortsätta lära sig och utvecklas inom maskininlärning och artificiell intelligens, eftersom det är ett snabbt växande område med stora möjligheter och utmaningar.
* Sammanfattningsvis, man kan använda maskininlärning för att lösa en klassificeringsuppgift på en verklig dataset. Resultaten kan användas för att välja den bäst presterande modellen och dess hyperparametrar för att använda i framtida tillämpningar.

# Teoretiska frågor

1. Kalle delar upp sin data i tre delar:

* Träning: Kalle lär sin modell att förstå sambandet mellan olika variabler och göra prediktioner.
* Validering: Kalle använder valideringsdelen för att utvärdera modellerna och välja den bästa modellen som fungerar bra på oberoende data.
* Test: Kalle testar den valda modellen i testdelen och får ett estimat av ”generalization error”. Testdelen används för att utvärdera den slutgiltiga prestandan av hur bra modellen fungerar.

1. Om Julia inte har skapat ett explicit ”validerings-dataset” för att utvärdera modellerna, kan hon använda en metod som kallas korsvalidering(cross-validation) för att välja vilken modell hon ska använda. Korsvalidering innebär att man delar upp träningsdatan i flera mindre delar (k-fold) och tränar modellerna på olika delar samtidigt som man testar dem på resten av datan.
2. Regressionsproblem är en typ av statistisk analys som syftar till att förstå sambandet mellan variabler. Det används för att prediktera eller förklara en beroende variabel baserat på oberoende variabler. Den beroende variabeln har kontinuerliga värden.

Några modeller som kan användas för att hantera regressionsproblem:

* Linjär Regression
* Support Vector Machines (SVM)
* Beslutsträd

1. Ett sätt att utvärdera hur väl en regressionsmodell passar en datamängd är att beräkna RMSE, vilket är ett mått som talar om för oss det genomsnittliga avståndet mellan de förutsagda värdena från modellen och de faktiska värdena i datasetet. Så kan vi tolka RMSE som prediktioners medelavstånd till de sanna värdena. Ju lägre RMSE, desto bättre kan en given modell "passa" en datauppsättning.

Formeln för att beräkna RMSE:

1. Klassificieringsproblem handlar om att tilldela objekt eller data till olika fördefinierade klasser eller kategorier.

I binär klassificering handlar det om att skilja mellan två möjliga utfall eller kategorier. Exempel: Att avgöra om en e-post är spam eller inte spam, eller om en patient har en viss sjukdom eller inte.

I multiklassklassificering handlar det om att kategorisera data i fler än två klasser eller kategorier där varje klasser kan ta fler kategorier samtidigt.

Exempel: {Brun/Ljusbrun/Mörkbrun, Soffa/Säng/Kudde, Mjuk/Mellan/Fast}

**Här är några exempel på modeller som används för klassificieringsproblem:**

* Linjär klassificering
* Logistisk regression
* Beslutsträd
* Support Vector Machines (SVM)
* K-nearest neighbors (KNN)
* Random Forest

**Tillämpningsområden:**

* Medicinsk diagnos: Sjuk eller Inte
* E-postfiltrering: Spam eller Inte spam
* Ansiktsigenkänning: Kvinna eller Man
* Finansiell bedrägeriupptäckt: Bedrägeri eller Inte

Confusion Matrix är en tabell som visar hur en klassificeringsmodell presterar genom att jämföra de faktiska och förutsagda klasserna.

**Den har fyra celler:**

* True Positive (TP): Antal korrekt klassificerade positiva exempel.
* False Negative (FN): Antal felaktigt klassificerade positiva exempel.
* False Positive (FP): Antal felaktigt klassificerade negativa exempel.
* True Negative (TN): Antal korrekt klassificerade negativa exempel.

1. K-Means algoritmen är en enkel algoritm som kan klustra denna typ av datauppsättning mycket snabbt och effektivt, ofta med bara några få iterationer. (Géron, 2019, s. 238) K-means använder iterativa tekniker för att dela upp datauppsättningar i kluster där varje kluster innehåller liknande datapunkter. Kluster är användbara för att utforska data, identifiera avvikelser och göra förutsägelser.

Exempel: Antag att vi har en stor datauppsättning med kundinformation, inklusive köphistorik och beteendemönster. Om man vill segmentera kunderna i olika grupper baserat på deras köpbeteende för att anpassa marknadsföring och erbjudanden. Med K-means-klustring man kan skapa uppsättningar av liknande kunder baserat på egenskaper.

När vi har skapat dessa kundsegment då kan vi skräddarsy marknadsföringsstrategier för varje grupp och rikta in dem med personliga erbjudanden, rekommendationer eller kampanjer.

1. Ordinal encoding används när man har en kategorisk variabel med en inbördes ordning mellan dess värden.

Exempel: Antag att man har en “Storlek på kläder” variabel med värdena “Liten”, “Medium” och “Stor”. Dessa värden har en inbyggd ordning och man omvandlar dessa värden till numeriska värden, t.ex. “Liten” = 1, “Medium” = 2 och “Stor” = 3.

One-hot encoding används för alla andra ML modeller när man har en kategorisk variabel utan någon inbördes ordning. Varje unikt värde i den kategoriska variabeln representeras som en separat binär variabel (0 eller 1).

Exempel: Om vi har en “Färg” variabel med värdena “Röd”, “Grön” och “Blå”, skapar vi tre nya variabler: “Röd”, “Grön” och “Blå”. Om en observation har färgen “Röd”, kommer “Röd” variabeln att vara 1 och de andra två variablerna kommer att vara 0.

Dummy variable encoding är en specifik form av one-hot encoding. Det innebär att man skapar en binär variabel för varje unikt värde i den kategoriska variabeln, men man använder bara N-1 variabler (där N är antalet unika värden).

Exempel: Om vi har en variabel "Blomma” med värdena “Ros”, “Tulpan” och “Dahlia”, skulle dummy variabel kodning skapa två nya kolumner: “Blomma\_Ros” och “Blomma\_Tulpan”. Om en observation är en ros, skulle “Blomma\_Ros” vara 1 och “Blomma\_Tulpan” skulle vara 0.

1. Jag tycker att båda har rätt. Göran beskriver två olika typer av data, "ordinal" och "nominal", medan Julia ger ett exempel på hur samma data kan tolkas på olika sätt beroende på sammanhanget. Ordinal data är data som kan rangordnas eller ordnas i en viss ordning, till exempel betygsskalan som VG-G-IG eller storlek av kläder som S-M-L. Nominal data är data som inte kan rangordnas, till exempel färger eller kön.

Julias exempel med färger visar hur samma data kan tolkas på olika sätt. Generellt sett är färger nominal data eftersom de inte kan rangordnas. Men i ett specifikt sammanhang, som att ha på sig en röd skjorta på en fest, kan färgen tolkas som ordinal data eftersom den ger en viss position eller status i gruppen.

1. Streamlit är ett ramverk med öppen källkod för att skapa dataapplikationer i Python för maskininlärnings och datavetenskapsteam. Det är utformat för att göra det enkelt för utvecklare att bygga och distribuera applikationer på ett enkelt och snabbt sätt. Streamlit kan användas för att:

* Skapa interaktiva dashboards för dataanalys
* Bygga och testa maskininlärningsmodeller
* Visualisera data på ett enkelt och effektivt sätt
* Presentera resultat och rapporter på ett interaktivt sätt
* Skapa prototyper för webbapplikationer

# Självutvärdering

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

Jag hade vissa svårigheter när jag förberedde mitt projekt eftersom maskininlärning inte är ett område som lätt kan läras in på kort tid. Förväntningarna på denna kurs var ganska höga. För att övervinna dessa utmaningar arbetade jag långa timmar och gjorde research för att förstå mitt arbete. Om jag skulle ge ett exempel; Det tog mig tid att bestämma vilken modell jag skulle använda och hur och att anpassa den till min rapportkodning. Det var så jag gjorde research i ämnet och hittade rätt information. Jag hoppas att det motsvarade förväntningarna.

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

Jag tycker att jag förtjänar att få ett G för även om jag gjorde mitt bästa så hade jag inte möjlighet att göra VG-delen.

1. Något du vill lyfta fram till Antonio?

Jag tycker att den kursen behöver mer tid och mer träning inför kunskapskontrollerna.

# Appendix A

* PCA är en effektiv metod för att extrahera de viktigaste funktionerna för att klassificera handskrivna siffror i MNIST-datasetet.
* Olika skalningsmetoder kan påverka modellernas prestanda. RobustScaler kan vara ett bättre val än StandardScaler om datamängden innehåller utliggare.

# Källförteckning

* [Hyperparameters of Random Forest Classifier - GeeksforGeeks](https://www.geeksforgeeks.org/hyperparameters-of-random-forest-classifier/)
* (Géron, 2019)
* [python - What is C parameter in sklearn Logistic Regression? - Stack Overflow](https://stackoverflow.com/questions/67513075/what-is-c-parameter-in-sklearn-logistic-regression)
* [Random forest - Wikipedia](https://en.wikipedia.org/wiki/Random_forest)
* [sklearn.ensemble.RandomForestClassifier — scikit-learn 1.4.1 documentation](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html)
* [machine\_learning\_ds23/l7/09\_ensemble\_learning\_random\_forest.pptx at main · AntonioPrgomet/machine\_learning\_ds23 · GitHub](https://github.com/AntonioPrgomet/machine_learning_ds23/blob/main/l7/09_ensemble_learning_random_forest.pptx)
* [sklearn.model\_selection.GridSearchCV — scikit-learn 1.4.1 documentation](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html)
* [machine\_learning\_ds23/l2/02\_klassificering.pptx at main · AntonioPrgomet/machine\_learning\_ds23 · GitHub](https://github.com/AntonioPrgomet/machine_learning_ds23/blob/main/l2/02_klassificering.pptx)
* [machine\_learning\_ds23/l2/03\_linreg\_logreg.pptx at main · AntonioPrgomet/machine\_learning\_ds23 · GitHub](https://github.com/AntonioPrgomet/machine_learning_ds23/blob/main/l2/03_linreg_logreg.pptx)