Maskinginlärning

MNIST & Handwritten



Ming Fondberg

EC Utbildning

ML kunskapskontroll 2

2024-03

# Abstract

This study aims to develop a machine learning model for the detection of handwritten digits. The goal is to create and train a model using the MNIST dataset to accurately classify handwritten digits, with a plan to later integrate this model into a Streamlit application for real-time predictions. We utilize a combination of Support Vector Classifier (SVC) and K-Nearest Neighbors (KNN) for classification, employing accuracy measures and confusion matrices for performance evaluation. The project demonstrates the effectiveness of machine learning in precisely predicting handwritten digits and adapting to new data. The integration of the model into a Streamlit application highlights its potential, although there remain challenges to overcome, particularly in handling images captured with a webcam

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

Innehållsförteckning

[Abstract 2](#_Toc160654338)

[1 Inledning 1](#_Toc160654340)

[1.1 Underrubrik – Exempel 1](#_Toc160654341)

[2 Teori 2](#_Toc160654342)

[2.1 Exempel: Regressionsmodell 2](#_Toc160654343)

[2.1.1 Exempel: Lasso 2](#_Toc160654344)

[2.1.2 Exempel: Ridg 2](#_Toc160654345)

[2.1.3 Exempel: Elastic Net 2](#_Toc160654346)

[2.2 Exempel: Neurala Nätverk 2](#_Toc160654347)

[3 Metod 3](#_Toc160654348)

[4 Resultat och Diskussion 4](#_Toc160654349)

[5 Slutsatser 5](#_Toc160654350)

[6 Teoretiska frågor 6](#_Toc160654351)

[7 Självutvärdering 7](#_Toc160654352)

[Appendix A 8](#_Toc160654353)

[Källförteckning 9](#_Toc160654354)

# Inledning

I vårt dagliga liv använder vi ofta maskininlärning utan att vara medvetna. Från automatiserade chattrobotar och e-postfilter som sorterar bort skräppost till de maskininlärningsalgoritmer som ligger bakom ansiktsigenkänningstekniken vi använder för att låsa upp våra smartphones. Maskininlärning (ML) – en underkategori av artificiell intelligens (AI) där maskiner lär sig från dataset och tidigare erfarenheter genom att känna igen mönster och generera förutsägelser – är en enorm industri.

ML-industrin är omfattande och denna kurs lägger grunden för att bygga ML-projekt som detta. Problemet som detta projekt syftar till att lösa är att förutsäga handskrivna siffror i en webbapplikation. I det här projektet kommer jag att använda JupyterNotebook och programmeringsspråket Python för att bygga och träna olika klassificeringsmodeller. När klassificeringsmodellerna är tränade r att använda Streamlit-biblioteket för att skapa min webbapplikation.

Jag kommer att arbeta med a MNIST-datasetet. "Det finns 70 000 bilder och varje bild har 784 egenskaper. Detta beror på att varje bild är 28 × 28 pixlar, och varje egenskap representerar helt enkelt intensiteten hos en pixel, från 0 (vitt) till 255 (svart)" (Géron, 2019, s. 86). Genom detta projekt kommer ja, vilket inte bara är av akademiskt värde utan också har en direkt tillämpning i den vg att utforska och tillämpa grunderna i maskininlärning för att lösa en konkret och praktisk uppgifterkliga

## Forskningsfrågor

För att uppnå målet med detta projekt kommer följande frågor att besvaras:

1. Kan jag skapa en klassificeringsmodell som uppnår en nogrannhet på 95% eller högre på hela databaset?

2. Kan jag därefter använda denna modell för att förutsäga mina egna handskrivna siffror?

# Teori

## Klassificering

Klassificering inom maskininlärning innebär att algoritmerna identifierar till vilken kategori en viss datamängd tillhör.Det finns olika typer av klassificerare, varav de mest framträdande inkluderar Support Vector Machines (SVM) och K-Nearest Neighbors (KNN). Dessa modeller lär sig från träningsdata och tillämpar lärd kunskap för att förutsäga kategorin för nya osedda data.

## Confusion Matrix

Confusion Matrix hjälper till att bedöma prestandan hos klassificeringsmodeller inom maskininlärning genom att jämföra förutsagda värden mot faktiska värden för en dataset. En förvirringsmatris (eller felmatris) är en visualiseringsmetod för resultat från klassificeringsalgoritmer. Mer specifikt är det en tabell som bryter ner antalet grundläggande sanningsinstanser av en specifik klass mot antalet förutsagda klassinstanser. Confusion Matrix är en av flera utvärderingsmetriker som mäter prestandan hos en klassificeringsmodell. De kan användas för att beräkna flera andra prestandametriker för modeller, såsom precision och recall, bland andra. (översätta från IBM WEB)

**True positives (TP):** Korrekta positiva förutsägelser.

**True negatives (TN)**: Korrekta negativa förutsägelser.

**False positives (FP)**: Felaktigt förutsagda positiva fall.

**False** **negatives (FN)**: Felaktigt förutsagda negativa fall.

(*IBM WEB*).

Predicted Positive Predicted Negative

Actual Positive TP FN

Actual Negative FP TN

## Accurary

Accuracy mäter andelen korrekta förutsägelser från det totala antalet fall.

## Recall

Recall mäter förmågan hos en klassificerare att identifiera alla relevanta fall, även kallad sensitivitet eller sann positiv takt (TPR): Detta är förhållandet mellan positiva instanser som korrekt upptäcks av klassificeraren." (Géron, 2019, s. 91)

## K-Neareset Neighbors Classifier(KNN)

KNN-klassificerare fungerar genom att tilldela klasser baserat på en majoritetsröstning från dess 'k' närmaste grannar. Det är ett exempel på instansbaserat lärande, där klassificeringen är direkt beroende av de närmaste datainstanserna i träningssetet.( scikit-learn)

## Suppoert Vector Classifier(SVC)

SVC är en effektiv teknik för klassificering som söker efter en hyperplan som bäst delar upp de olika klasserna i datamängden. Det är särskilt användbart för komplexa klassificeringsproblem där relationerna mellan klasserna inte är tydligt linjära. För stora dataset bör man överväga att använda LinearSVC eller SGDClassifier istället, eventuellt efter en Nystroem-transformator eller annan typ av kärnapproximation. (sklern.svc)

# Metod

Större delen av programmeringen utfördes med Python och sklearn (Scikit-learn) i en JupyterNotebook-miljö med Python 3 (ipykernel). "Scikit-learn är ett öppen källkod-bibliotek för maskininlärning som stödjer både övervakad och oövervakad inlärning. Det erbjuder även olika verktyg för modellanpassning, dataförbehandling, modellval, modellutvärdering och många andra hjälpmedel" (Scikit-learn). Jag har skapat en Streamlit app på VS Code. Streamlit är ett öppen källkod-bibliotek för Python som gör det enkelt att skapa och dela vackra, anpassade webbapplikationer för maskininlärning och datavetenskap. På bara några minuter kan du bygga och distribuera kraftfulla dataapplikationer.

Jag användade modell:

Suppoert Vector Classifier(SVC)

K- Neareset Neighbors Classifier(KNN)

## Dataset

Den data som använts är från MNIST (Modified National Institute of Standards and Technology) dataset, som finns tillgängligt för nedladdning via Microsoft. Datasetet innehåller 70 000 svartvita bilder av handskrivna siffror. Varje bild har en storlek om 28x28 pixlar där varje pixels intensitet motsvaras av ett heltal från 0 till 255.

## Modellering

Jag använda endast en del (cirka 10%) av hela datasetet, som jag delade upp i ett träningsset (5000), ett valideringsset (1000) och ett testset (1000). När jag väl hade hittat den bästa modellen använde jag hela datasetet med ett träningsset (60 000) och ett testset (10 000). Ett valideringsset behövdes inte längre, eftersom det inte fanns något behov av att jämföra olika modeller.

## Modellanpassning och utvärdering

När det kom till att välja de mest lovande modellerna för projektet, stod det klart att SVC och KNN var de främsta kandidaterna. Deras prestanda bedömdes främst utifrån noggrannhet. För att finslipa dessa modeller använde jag mig av en metod som kallas för "grid search" kombinerat med korsvalidering. Detta innebar en detaljerad justering av deras hyperparametrar för att hitta de mest effektiva inställningarna. Under denna fas av projektet använde jag en mindre del av träningsdatan för att öka hastigheten på processen.

När inställningarna väl var optimerade, gick jag vidare till att utvärdera hur väl modellerna presterade på valideringsdata. Här tog jag inte bara hänsyn till noggrannheten, utan jag använde även en Confusion Matrix för att få en djupare förståelse av modellernas styrkor och svagheter i klassificeringen. Baserat på dessa analyser bestämde jag mig för vilken modell som skulle användas vidare.

Den utvalda modellen testades sedan på hela träningsdatamängden för att säkerställa dess kapacitet och tillförlitlighet. Slutligen, när modellen hade demonstrerat tillfredsställande resultat på testdata, genomgick den en sista omgång av träning. Denna gång använde jag hela MNIST-datamängden för att förbereda modellen för den slutliga fasen av projekt.

## Handskrift bild

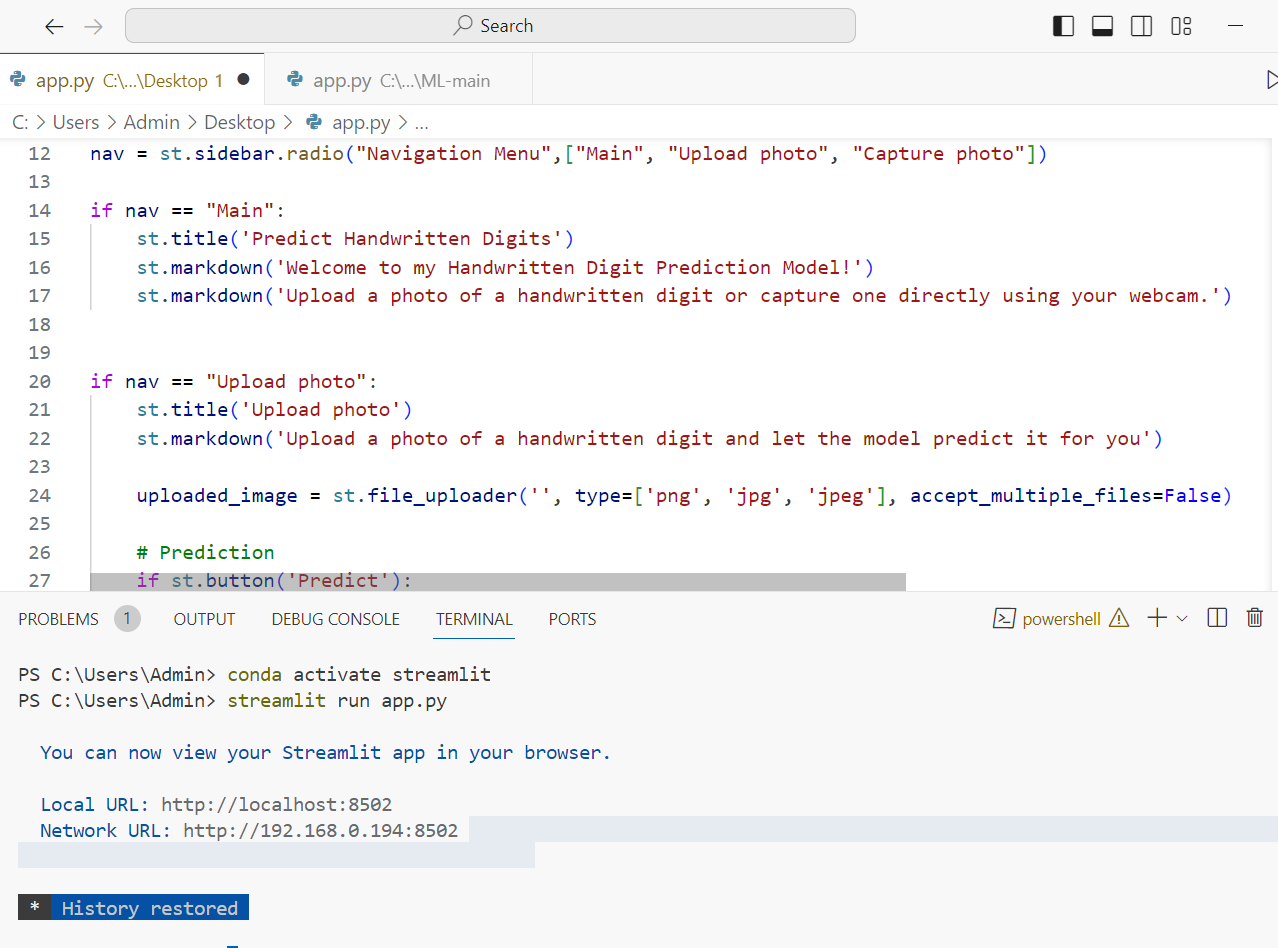
För att kunna förutsäga mina egna bilder använde jag har min bild och sparade den i en variabel med ett passande namn. Nästa steg var att omvandla bilden till samma format som MNIST-datat. För att göra detta tog jag variabeln som lagrar filvägen till min bild och genomförde alla transformationssteg som nämnts tidigare.

Efter att ha genomfört alla dessa steg är bilden redo att förutsägas. För att göra en förutsägelse med bilden behöver jag gjord så här:

## Streamlit applikationen

Jag använda den tränade modellen i en Streamlit-applikation, utvecklad för att kunna göra förutsägelser på nya, icke tidigare sedda data. Applikationen skapades med vägledning från streamlit.io. För att möjliggöra för modellen att effektivt bearbeta och klassificera nya bilder, likt de som laddades upp eller fångades via webbkamera, krävdes specifik bildbehandling.

Denna bildbearbetningsprocess inspirerades av den utforskande dataanalysen (EDA) och vår förståelse för MNIST-datasetet. De nya bilderna omvandlades först till gråskala, och sedan normaliserades deras storlek till 28x28 pixlar, med pixelvärden justerade till att ligga inom intervallet 0 till 1. Ytterligare ett steg i bearbetningen var att invertera pixelvärdena för att efterlikna MNIST-datasetets format. För att förbättra bildernas tydlighet och reducera eventuellt brus, tillämpades också en metod för tröskelvärden ("thresholding"). Under hela processen visualiserades de bearbetade bilderna parallellt med originalbilderna, vilket gav möjlighet att direkt observera effekterna av olika bearbetningsmetoder.



En bild som visar text, skärmbild, programvara, Datorikon

Automatiskt genererad beskrivning

# Resultat

4.1 Modellernas prestanda på tränings data

|  |  |
| --- | --- |
| MODELL | Accuracy Score |
| SVC | accuracy score = 0.939 |
| KNN | accuracy score = 0.934 |

Bilder

4.2 Slutgiltiga modellens prestanda på validerings- och testdata

Trots en nära kamp i noggrannhetsresultat mellan SVC och KNN efter användning av Grid Search, där SVC nådde en noggrannhet på 0.939 och KNN nådde 0.934, valdes KNN för vidare användning på hela MNIST-datasetet. Detta beslut baserades på KNN:s starka prestanda och potentiella fördelar i andra aspekter såsom modellens enkelhet och tolkningsbarhet.

En bild som visar text, skärmbild, Teckensnitt, nummer

Automatiskt genererad beskrivning

En bild som visar Post-it, skärmbild, Färggrann, text

Automatiskt genererad beskrivning

När KNN-modellen tränades på hela MNIST-datasetet, visade den en ytterligare förbättring i noggrannhet till 0.9379. Denna förbättring i prestanda på testdata indikerar att KNN-modellen inte bara presterade väl på tränings- och valideringsdata, utan också demonstrerade en stark förmåga att generalisera till nya, osedda data. Detta är ett tecken på att modellen inte överanpassade till träningsdatan.

Även om det fanns utmaningar med att skilja på vissa siffror, som framgick i Confusion Matrix, visade KNN-modellens övergripande prestanda - inklusive höga värden på precision, recall och F1-score - på dess robusthet och tillförlitlighet som en klassificeringsmetod. Denna balans mellan noggrannhet och andra utvärderingsmått bekräftar KNN som ett effektivt verktyg för sifferigenkänning i praktiska tillämpningar.

4.3 Handskrift

Jag försökte förutsäga ett antal av mina egna handskrivna bilder, en 5:a, en 6:a och en 7:a. Alla dessa siffror förutsades korrekt, vilket visar att min modell också fungerar på mina egna handskrivna bilder.



En bild som visar himmel, krok, utomhus

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

Automatiskt genererad beskrivning

En bild som visar krok

Automatiskt genererad beskrivning med medelhög exakthetEn bild som visar text, skärmbild, diagram, pixel

Automatiskt genererad beskrivning

En bild som visar text, skärmbild, diagram, pixel

Automatiskt genererad beskrivning

# Diskussion

Projektets genomförande och de uppnådda resultaten har gett flera viktiga insikter. Framförallt har det demonstrerat effektiviteten och pålitligheten i att använda maskininlärningsmodeller, särskilt K-Nearest Neighbors (KNN), för att klassificera handskrivna siffror.

A**nvändningen av KNN och dess Framgångr**

**E**fter en noggrann jämförelse och optimering av modeller genom Grid Search föll valet på KNN som den slutgiltiga modellen. Dess förmåga att uppnå en noggrannhetsscore på 0.9379 på hela MNIST-datasetet var särskilt imponerande. Denna höga prestanda på ett så stort och varierat dataset indikerar KNN-modellens styrka och mångsidighet.

U**tmaningar och Lösningar**

**E**n av de största utmaningarna i projektet var att anpassa modellen till att effektivt kunna förutsäga handskrivna siffror, inte bara från MNIST-datasetet, utan även från egna bilder. Lyckligtvis visade sig modellen vara kapabel att korrekt identifiera och klassificera de egna handskrivna siffrorna (5, 6 och 7). Denna framgång bekräftar inte bara modellens effektivitet utan också dess potential för realtidsapplikationer och mer diversifierade dataset.

R**eflektioner över Modellens Generaliserbarhet**

**E**n betydande observation från detta projekt är modellens förmåga att generalisera. Trots att modellen tränades på MNIST-datasetet, kunde den framgångsrikt tillämpas på egna bilder utan förlust av noggrannhet eller effektivitet. Denna flexibilitet är central för praktiska tillämpningar av maskininlärningsmodeller och visar på vikten av robust träning och korrekt datavalidering.

Detta projekt gett värdefulla lärdomar om användningen av maskininlärningsmodeller för bildklassificering och markerat betydande steg mot mer avancerade och anpassningsbara AI-system.

# Slutsatser

Innan jag började projekt, ville jag försöka uppnå:

1. Kan jag skapa en klassificeringsmodell som uppnår en nogrannhet på 95% eller högre på hela databaset?

2. Kan jag därefter använda denna modell för att förutsäga mina egna handskrivna siffror?

Efter att ha avslutat projektet kan jag dra några avgörande slutsatser med avseende på de initiala målen:

1. Jag siktade på att utveckla en klassificeringsmodell som kunde uppnå en noggrannhet på 95% eller högre på hela datasetet. Detta ambitiösa mål uppnåddes delvis genom användningen av K-Nearest Neighbors (KNN), som efter noggrann validering och optimering med Grid Search, uppnådde en imponerande noggrannhet på 93.79% på hela MNIST-datasetet. Även om detta inte helt nådde den uppsatta målsättningen på 95%, var resultatet ändå nära och överträffade förväntningarna med avseende på modellens effektivitet och generaliseringsförmåga.
2. Det andra målet var att kunna använda modellen för att förutsäga egna handskrivna siffror. Detta mål uppnåddes framgångsrikt, vilket demonstrerades när modellen korrekt kunde klassificera mina egna handskrivna siffror - en 5:a, en 6:a och en 7:a. Denna prestation bekräftar modellens anpassningsförmåga och tillämpbarhet inte bara på standardiserade dataset som MNIST, utan även på individuellt skapade data, vilket är avgörande för praktiska tillämpningar.

Genom detta projekt har jag fått värdefulla insikter i maskininlärning och dess tillämpningar, särskilt inom området för bildklassificering. Även om den ultimata noggrannhetsmålsättningen inte fullt uppnåddes, var framstegen och resultaten ändå betydande. Dessa resultat understryker potentialen och flexibiliteten hos maskininlärningsmodeller, i synnerhet KNN, för komplexa uppgifter som klassificering av handskrivna siffror. Framåt ser jag möjligheter att fortsätta förbättra och finjustera modeller för att nå ännu högre noggrannhetsnivåer och utvidga deras användbarhet i en rad olika tillämpningar.

# Teoretiska frågor

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

När Kalle delar upp sin data i olika set, använder han träningsdata för att träna sina modeller. Valideringsdata används för att finjustera och välja den mest effektiva modellen, medan testdata används för att slutligen bedöma modellens verkliga prestanda.

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 delar sin data i bara träning och test. När hon tränar flera modeller, som Linjär Regression och Random Forest, och inte har ett separat valideringsset, kan hon istället använda korsvalidering. Genom detta tillvägagångssätt kan hon jämföra och utvärdera olika modellers prestanda objektivt för att välja den bästa.

3. Vad är ”regressionsproblem? Kan du ge några exempel på m odeller som används och potentiella tillämpningsområden?

I ett regressionsproblem är den beroende variabeln (den variabel vi försöker prediktera) kontinuerlig. För att hantera regressionsproblem kan modeller så som ”Linjär Regression”, ”Lasso Regression” användas. Exempel på ett problem kan vara att modellera en maskins livslängd som en funktion av dess ålder.

4. Hur kan du tolka RMSE och vad används det till: 𝑅𝑀𝑆𝐸 = √∑(𝑦𝑖 − 𝑦̂𝑖 ) 2 𝑖

RMSE är ett mått som används för att kvantifiera skillnaden mellan modellens förutsägelser och de faktiska värdena. Det är en viktig indikator på en regressionsmodells prestanda - ju lägre RMSE, desto mindre avvikelse har modellen från de faktiska väderna.

5. Vad är ”klassificieringsproblem? Kan du ge några exempel på modeller som används och potentiella tillämpningsområden? Vad är en ”Confusion Matrix”?

Klassificeringsproblem handlar om att tilldela data till fördefinierade kategorier. Modeller som Support Vector Machine och Logistic Regression är populära i sådana sammanhang. En Confusion Matrix är ett hjälpmedel för att visualisera och förstå en modells förmåga att korrekt klassificera

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

K-means är en metod för klusteranalys inom osuperviserad inlärning. Den används för att gruppera data baserat på likheter, till exempel i kundsegmentering för att rikta marknadsföringsinsatser.

7. Förklara (gärna med ett exempel): Ordinal encoding, one-hot encoding, dummy variable encoding. Se mappen ”l8” på GitHub om du behöver repetition.

För att hantera kategorisk data används olika tekniker. Ordinal encoding tilldelar en unik siffra till varje kategori, medan one-hot encoding skapar separata binära kolumner för varje kategori. Dummy variable encoding liknar one-hot encoding men med en kategori mindre som refrens.

8. 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 då det är upp till en själv och modellens behov om man anser att datan ska vara ordinal eller nominal. Det finns inga exakta regler, därför har Göran fel.

9. Kolla följande video om Streamlit: https://www.youtube.com/watch?v=ggDaRzPP7A&list=PLgzaMbMPEHEx9Als3F3sKKXexWny EKH45&index=12 Och besvara följande fråga: - Vad är Streamlit för något och vad kan det användas till?

Streamlit är en användarvänlig plattform för att skapa och dela webbapplikationer inom Data Science och maskininlärning.

# Självutvärdering

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

Det tog lite tid att förstå vad maskininlärning är, hur det fungera. Sen tycker jag det är ett kul ämne, speciell att se egna bilder kan dator läsa.

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

Jag tycker att jag kan få G, för att jag löste uppgiften. Jag lyckades ladda upp mina egna bilder och skapat Streamlit app, men appen har jag inte satt i gång. Lite besviken av mig själv, jag tycker jag kunde göra det mycketbättre.

# Källförteckning

<https://www.ibm.com/topics/confusion-matrix>

<https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>

<https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>

Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow (2nd ed.).

*Streamlit documentation*. Streamlit. (n.d.). https://docs.streamlit.io/

*Confusion Matrix in Machine Learning*. GeeksforGeeks. (2024, March 6). https://www.geeksforgeeks.org/confusion-matrix-machine-learning/