Kunskapskontroll

Maskininlärningsprojekt

Tova Thorén

EC Utbildning

Kunskapskontroll

202403

Abstract

This project focuses on implementing a machine learning model for handwritten digit detection. The purpose is to train a model on the MNIST dataset to classify images of handwritten digits with high accuracy, to later integrate the model into a Streamlit application for real-time predictions. Through the utilization of a Support Vector Machine (SVM) classifier and evaluation metrics like accuracy and Confusion Matrices, the project demonstrates the efficacy of machine learning in predicting handwritten digits accurately and generalizing to new data. The integration of the model into a Streamlit application further showcases its potential, albeit with room for improvement in handling images taken via webcam, due to challenges such as lighting variations.

Innehållsförteckning

1 Inledning

Bildigenkänning är ett framstående område inom datavetenskap och artificiell intelligens. Tekniken bakom bildigenkänning möjliggör för datorer att tolka visuell information från digitala bilder, vilket blir alltmer betydelsefullt i en värld där samhället i stor utsträckning digitaliseras. En central del av bildigenkänning är sifferigenkänning, där målet är att identifiera och klassificera handskrivna siffror från digitala bilder (Demirkaya & Çavuşoğlu, 2022).

I dagens digitaliserade samhälle hanteras och lagras en stor mängd av all information elektroniskt. Därigenom får tekniker för att känna igen handskrivna siffror en stor betydelse, för att effektivt kunna integrera analog information i de digitala systemen som används. Praktiska tillämpningar för en sådan teknik är exempelvis inom postsortering, bearbetning av bankcheckar och tolkning av handskrivna formulär (Demirkaya & Çavuşoğlu, 2022).

Ett av de mest använda dataseten för att utforska sifferigenkänning är MNIST. Datasetet består av en omfattande samling bilder på handskrivna siffror och refereras ofta till som ”hello world” av maskininlärning (Géron, 2019).

1.1 Syfte

Syftet med detta projekt är följaktligen att träna en modell att klassificera bilder av handskrivna siffror och prediktera rätt siffra på given data, med hjälp av maskininlärning. Modellen kommer sedan att integreras i en Streamlit-applikation för att göra prediktioner på ny data.

För att uppnå syftet ska följande två frågeställningar besvaras:

- Kan en maskininlärningsmodell tränas för att prediktera handskrivna siffror från MNIST-dataset med en noggrannhet på minst 90%?
- Kan en modell tränad på MNIST-datasetet generalisera till ny data via en Streamlit-applikation?

2 Teori

Sifferigenkänning är ett klassificeringsproblem. Med hjälp av en maskininlärningsalgoritm klassificeras bilderna och siffran på bilden predikteras tillhöra en av klasserna mellan 0-9. I följande avsnitt beskrivs de teoretiska grunder som maskininlärningsalgoritmerna bygger på, samt hur modellernas prestation och predikteringsförmåga utvärderas.

2.1 Klassificerare

Klassificeringsproblem tillfaller subkategorin övervakad maskininlärning (”supervised learning”), och innebär att den beroende variabeln (den variabel vi försöker prediktera) är av kategorisk karaktär. Det finns en rad olika maskininlärningsalgoritmer för att hantera klassificeringsproblem (Géron, 2019). I det aktuella projektet används huvudsakligen modellerna SVM klassificerare och Random Forest, vilka beskrivs närmre nedan.

2.1.1 Support Vector Machines (SVM)

Support Vector Machines (SVM) är en linjär klassificerare som kan hantera såväl linjär som icke-linjär separerbar data. SVM klassificering bygger på konceptet ”maximal marginalklassificering”, vilket handlar om att dela upp datapunkterna i två linjärt separerbara grupper med så stor marginal som möjligt. Men modellen hanterar även icke-linjär separerbar data genom att transformera den till högre dimensioner med hjälp av så kallade ”kernel-tricks”. Detta gör modellen till en robust klassificerare och gör att den fungerar bra även på komplexa och högdimensionella dataset. Det är dock viktigt att notera att SVM modeller generellt är känsliga för skalan på data, vilket innebär att det är vanligt att standardisera data för att säkerställa att modellen presterar så bra som möjligt (Géron, 2019).

2.1.2 Random Forest (RF)

Random Forest (RF) är en annan kraftfull maskininlärningsmodell som likt SVM kan användas för icke-linjär separerbar data. Men till skillnad från SVM, som bygger på en enda modell, använder RF en teknik som kallas ”ensemble learning”. Detta innebär att modellen bygger flera beslutsträd och kombinerar deras individuella prediktioner för att göra en slutgiltig klassificering. En stor fördel med RF är dess robusthet mot olika typer av data, inklusive data med olika skalor. Det beror på att RF baserar sina beslut på relativ ordning av ”features” värden snarare än deras absoluta värden. Detta gör att modellen blir mindre känslig för skalan på data och gör det möjligt att använda features i olika enheter eller skalor utan att behöva förhandsbehandla dem. Random Forest utför implicit variabelselektion genom att välja ett slumpmässigt set av variabler för att bygga varje beslutsträd, vilket gör modellen lämplig för högdimensionella dataset (Géron, 2019).

2.2 Utvärderingsmått

För att kunna bedöma en modells förmåga att generalisera väl på ny data, krävs ändamålsriktiga mätetal. Valet av utvärderingsmått bör huvudsakligen styras av det specifika problemet och dess mål.

2.2.1 Confusion Matrix

Resultatet av en klassificeras prediktioner kan sammanfattas i en ”Confusion Matrix”, vilket är ett verktyg som visuellt tydliggör hur modellen har klassificerat given data. Nedan är en enkel demonstration av en sådan tabell (vid binära klassificeringsproblem).

	Predikterat Positiv	Predikterat Negativ
Faktiskt Positiv	Sanna Positiva (TP)	Falska Negativa (FN)
Faktiskt Negativ	Falska Positiva (FP)	Sanna Negativa (TN)

Utifrån en ”Confusion Matrix” kan följande kvantitativa mått beräknas.

2.2.2 Precision

Mäter andelen av de positivt predikterade fall som faktiskt är positiva.

$$Precision = \frac{TP}{TP + FN}$$

2.2.3 Träffsäkerhet (”recall”)

Mäter andelen av den positiva klassen som vi faktiskt predikterar korrekt.

$$Recall = \frac{TP}{TP + FN}$$

2.2.4 F1-score

För att ha ett balanserat mätetal som tar hänsyn till både precision och recall så används F1-värdet.

$$F1 = \frac{1}{\frac{1}{P} + \frac{1}{R}}$$

2.2.5 Noggrannhet (”accuracy”)

Mäter andelen korrekta prediktioner i relation till det totala antalet prediktioner. Sträcker sig från 0 (minst exakt) till 1 (mest exakt).

$$Accuracy = \frac{(TP+TN)}{n}$$

2.3 Korsvalidering

För att säkerställa en tillförlitlig utvärdering av modellens prestanda är det vanligt att dela upp data i tre separata set: träningsset, valideringsset och testset. Denna uppdelning möjliggör att modellen tränas på träningsdatan, optimeras med hjälp av valideringsdatan och slutligen utvärderas på testdatan för att bedöma dess prestanda på ny, oberoende data. Syftet med denna strategi är att minimera risken för överanpassning av modellen till träningsdata (Yang, 2021).

Ett alternativ till att använda separata dataset är att använda sig av korsvalidering. Vid korsvalidering delas datamängden in i k mindre delar, så kallade ”folds”. Sedan tränas och utvärderas modellen k gånger, där vid varje fold används k-1 data som valideringsset och de återstående som träningsset. Detta utgör en mer robust utvärdering av modellens prestanda eftersom varje observation har möjlighet att ingå i både tränings- och valideringsset under olika iterationer (Géron, 2019).

3 Metod

I följande avsnitt beskrivs de steg som har tagits för att nå resultatet, inkluderat den data som använts och hur den förberetts inför maskininlärning, val av modell, anpassning av modellen och slutligen hur modellernas prestanda har utvärderats. Dessutom redovisas tillvägagångssättet med Streamlit-applikationen, med fokus på hur bilderna behandlats för att vara kompatibla med den tränade modellen.

3.1 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.

3.2 Explorativ dataanalys (EDA)

Efter att ha hämtat och laddat in datasetet genomfördes en explorativ dataanalys (EDA) för att få en djupare förståelse för datan. Detta inkluderade att visualisera de fem första bilderna i datasetet tillsammans med deras tillhörande labels.

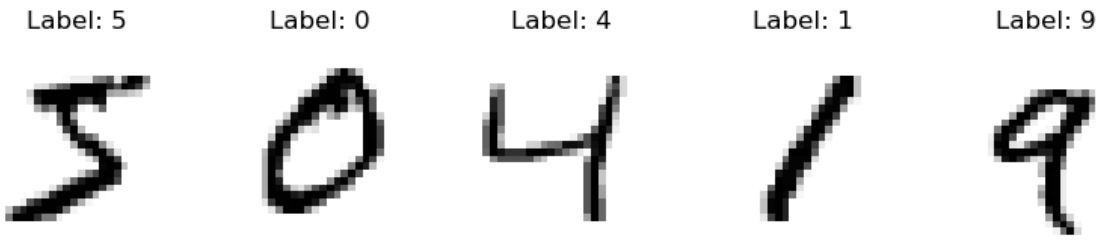


Bild 1: En visualisering av de första fem bilderna i datasetet.

Under EDA framkom även att datasetet var relativt balanserat, dvs det är en jämn fördelning av siffrornas/klassernas representation i datasetet. Detta påverkade mitt val av utvärderingsmått, eftersom det innebär att noggrannhet kan användas som mått på modellernas prestanda.

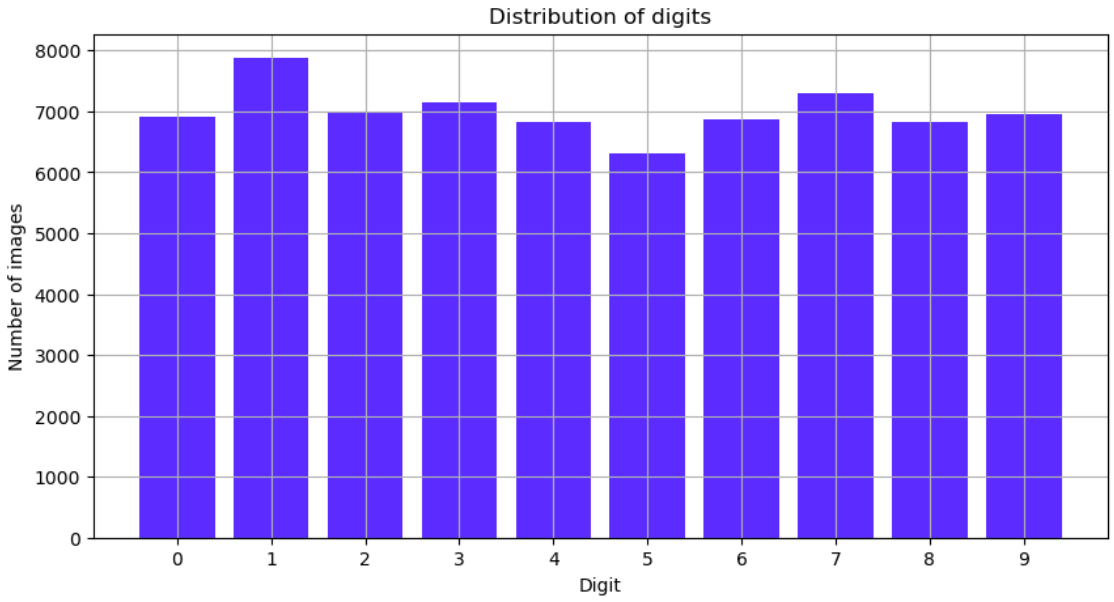


Bild 2: Distributionen av klasser i datasetet.

Vidare bidrar analysen även med insikter kring struktur och form på data samt hur pixlarnas intensitet är kodade (inverterade), vilket kommer vara viktig information när det är dags att formatera nya bilder som modellen ska tolka och prediktera.

3.3 Förberedning av data

Då datasetet visade sig vara välbalanserat och lämpligt för vidare bearbetning, genomfördes endast en standardisering av datan för att förbereda den inför maskininlärningsalgoritmerna. Innan datan standardiserades delas dock den totala datamängden på 70 000 bilder in i tre separata set; träningsdata (50 000), valideringsdata (10 000) och testdata (10 000), varav de senare kommer användas för att anpassa modellerna och utvärdera modellernas prestanda.

3.4 Modellval

Valet av modell baserades på problemets natur och datasetets storlek och struktur. Eftersom projektet fokuserade på klassificering av bilder från MNIST-datasetet, valdes klassificeringsmodeller. Det var även viktigt att modellen skulle kunna hantera högdimensionell data, då MNIST-dataset består av bilder om 784 pixlar vardera (28x28). Modellerna SVC och Random Forest valdes därmed på teoretisk grund, utifrån deras förmåga att hantera icke-linjära relationer och effektivitet vid högdimensionella dataset. Initialt utforskades dock ett antal olika klassificerare, för att få en grov uppskattning på hur väl olika modeller presterade på datasetet. I detta steg tränades klassificerarna med de fördefinierade standardvärdena för hyperparametrarna på en mindre mängd av träningsdatan och utvärderades sedan på hela träningsdatamängden.

3.5 Modellanpassning och utvärdering

Efter att ha valt de mest lovande modellerna (SVC och Random Forest) med noggrannhet som utvärderingsmått på deras prestanda, anpassades modellerna genom att optimera deras hyperparametrar med hjälp av ”gridsearch” och korsvalidering. Även i detta steg användes en mindre mängd av träningsdatan, för att snabba på träningen. Därefter utvärderades modellernas prestanda på valideringsdata, återigen med avseende på noggrannhet men även genom en ”Confusion Matrix”. Utifrån resultaten valdes den slutgiltiga modellen för vidare träning på hela träningsdatamängden och utvärdering på testdata. Efter att ha uppnått ett tillfredsställande resultat vid utvärdering på testdata så förbereddes modellen slutligen genom att tränas om på hela datamängden i MNIST, inför den sista delen i projektet.

3.6 Streamlit applikationen

I den sista delen av projektet integrerades modellen i en Streamlit-applikation för att prediktera siffror på ny, utomstående data. Applikationen utvecklades i enlighet med anvisningar från läraren och med hjälp av tillgänglig dokumentation på Internet. För att modellen skulle kunna hantera de nya bilder som laddades upp eller togs med hjälp av webbkameran, behövde bilderna formateras på ett sätt som liknade MNIST-datasetet som modellen tränats på.

Genom att använda insikterna från EDA och förståelsen för MNIST datasetet, bearbetades de nya bilderna bland annat genom att konverteras till gråskala, standardiseras till en storlek på 28x28 pixlar och normaliseras så att pixelvärdena låg inom intervallet 0 till 1. Därefter inverterades pixelvärdena för att matcha MNIST. För att förtydliga siffrorna i bilderna och minska eventuellt brus användes även en enklare ”thresholding” teknik. De behandlade bilderna visualiserades bredvid originalbilderna för att under arbetets gång kunna se hur de förändrades med olika metoder.

Nedan följer en skärmdump över den kod som använts för att bearbeta bilderna och göra nya prediktioner i Streamlit-applikationen.

```python
def preprocess_image(image):
    # Convert to grayscale
    gray_img = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Resize to 28x28 pixels and normalize pixel values [0,1]
    resized_img = cv2.resize(gray_img, (28, 28))
    normalized_img = resized_img / 255.0

    # Invert pixel vaues due to MNIST data being inverted
    inverted_img = (1 - normalized_img) * 255.0

    # Apply thresholding, setting pixels below mean to black
    mean_pixel_value = np.mean(inverted_img)
    inverted_img[inverted_img < mean_pixel_value] = 0

    # Reshape, 1d-array and then 2d-array (1, 784)
    flattened_img = inverted_img.flatten()
    reshaped_img_2d = flattened_img.reshape(1, -1)

    # Load and apply scaler
    scaler = joblib.load("scaler.pkl")
    img_ready = scaler.transform(reshaped_img_2d)

    return img_ready.flatten()

def make_prediction(image):
    model = joblib.load("final_model.pkl")
    predicted_number = model.predict([image])

    return predicted_number[0]
```

Bild 3. Utdrag från kod som använts för att bearbeta bilderna och göra nya prediktioner i Streamlit.

4 Resultat

I följande avsnitt presenteras resultaten från modellernas prestanda på MNIST-datasetet samt exempel på hur modellen presterat på Streamlit applikationen.

4.1 Modellernas prestanda på tränings- och valideringsdata

I tabellen nedan redovisas resultaten från den initiala träningen av klassificerare och deras prestanda på träningsdatan.

Modell	Accuracy (träningsdata)
Linear SVC	0.83
Logistic Regression	0.90
SVC	0.93
k-Nearest Neighbors	0.89
Random Forest	0.94

Modellerna SVC och Random Forest presterade bäst under det första urvalet, sett utifrån deras mått av noggrannhet (0.93 resp. 0.94). Nedan redovisas de mest optimala hyperparameterinställningarna för respektive modell och deras prestanda på valideringsdata.

Modell	Hyperparametrar (gridsearch)	Accuracy (valideringsdata)
SVC	{'C': 0.5, 'gamma': 1, 'kernel': 'poly'}	0.9411
Random Forest	{'max_depth': 20, 'n_estimators': 200}	0.9376

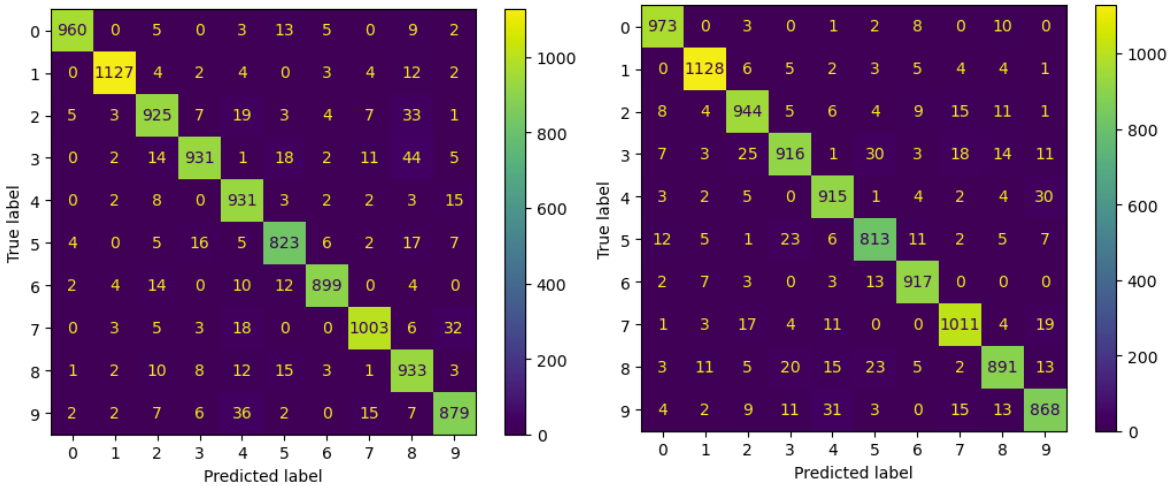


Bild 4. Modellernas prestation på valideringsdata, enligt en ”Confusion Matrix”. SVC modellen är placerad till vänster i bild.

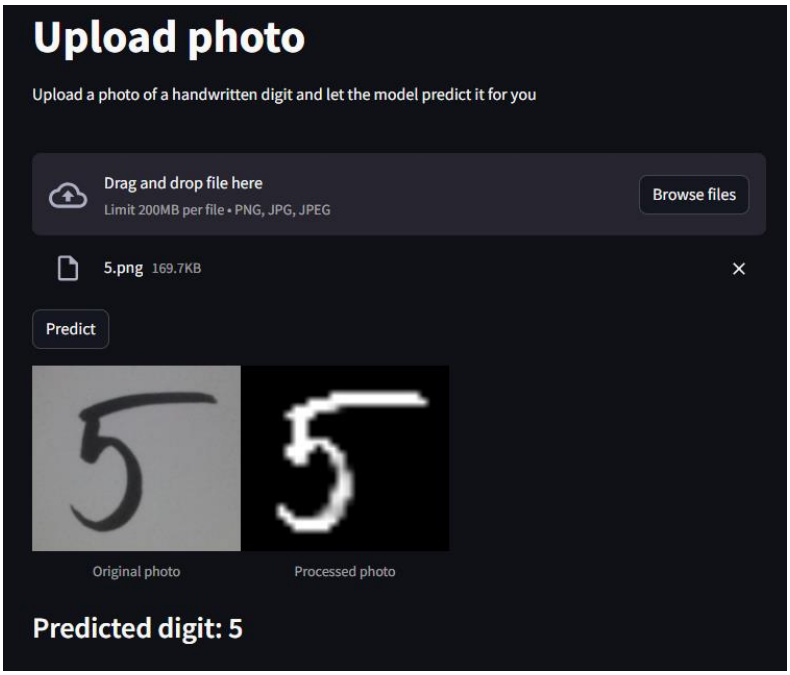
4.2 Den slutgiltiga modellens prestanda på validerings- och testdata

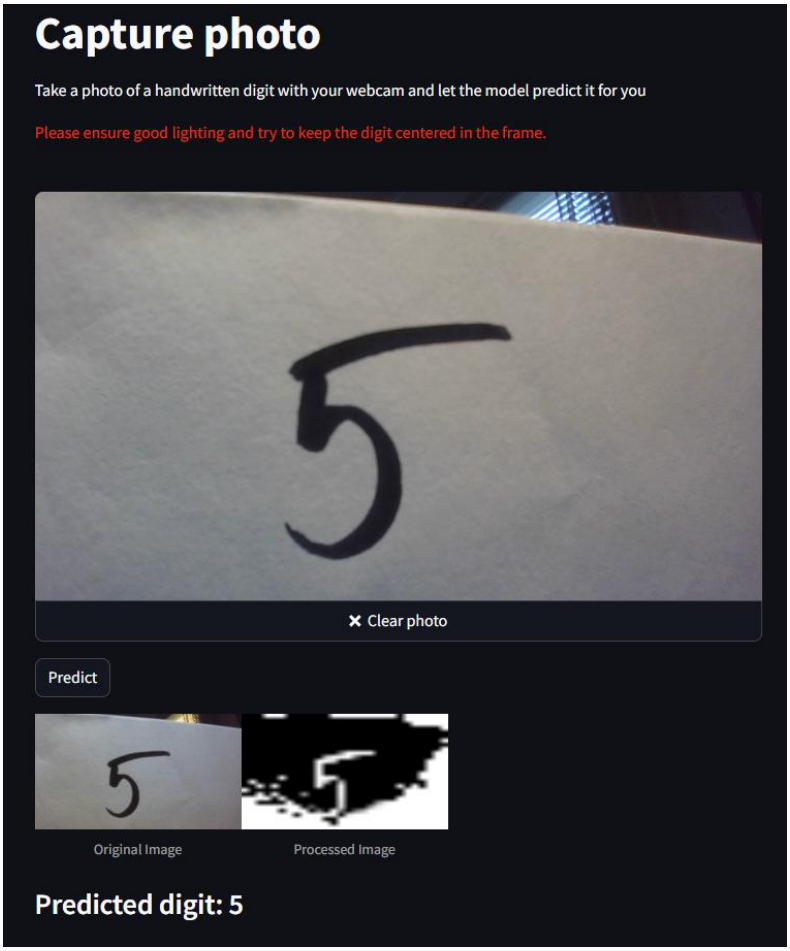
Trots svårigheter med att skilja på vissa siffror (som framgår i ”Confusion Matrix”) visade sig SVC modellen vara den mest lämpliga baserat på dess noggrannhet. Efter att ha tränats om med de optimala hyperparameterinställningarna på hela träningsdatan, uppnådde modellen en noggrannhet på 0.98 vid utvärdering på valideringsdata. Vid den slutgiltiga utvärderingen på testdata uppnådde modellen återigen en noggrannhet på 0.98. Denna prestanda tolkas som att modellen generaliserar väl till ny data och inte överanpassar till träningsdatan. Modellen uppnådde även höga värden på måtten ”precision”, ”recall” och ”f1-score”.

4.3 Resultat från Streamlit applikationen

Modellen predikterar rätt i samtliga fall vid uppladdning av bilder till Streamlit applikationen, men lite sämre när bilder tas direkt med webbkameran.

Nedan följer två exempel på prediktioner som modellen gör.





5 Diskussion

I följande avsnitt diskuteras resultaten i förhållande till projektets syfte, inklusive eventuella metodvals inverkan på resultatet. Vidare diskuteras även resultaten från Streamlit-applikationen och potentiella åtgärder för att förbättra modellens förutsättningar i framtiden.

5.1 Frågeställning 1: Kan en maskininlärningsmodell tränas för att prediktera handskrivna siffror från MNIST-dataset med en noggrannhet på minst 90%?

Den första delen i projektet handlade om att modellera MNIST-data och undersöka möjligheten att använda maskininlärning för att prediktera handskrivna siffror med hög noggrannhet. Resultatet visar på effektiviteten hos maskininlärningsmodeller, så som SVC och Random Forest, för den specifika uppgiften. Den slutgiltiga modellen (SVC) uppvisade hög noggrannhet på både validerings- och testdata, vilket även indikerar på modellens förmåga att generalisera till ny data.

Det tåls dock att betona risken med att träna modeller på en begränsad mängd träningsdata. Beslutet att använda en mindre datamängd för träning gjordes för att minska beräkningstiden och påskynda träningsprocessen. Detta kan dock ha påverkat beräkningen av optimala hyperparametrar, då modellerna inte tränades på hela datamängden. Trots detta visar resultaten att SVC-modellen presterade väl både på den begränsade träningsmängden och när den tränades om på hela träningsdatamängden med de valda hyperparametrarna. Samtidigt presterade modellen bättre när den tränades om på hela träningsdatamängden, vilket innebär att Random Forest modellen troligtvis också hade presterat bättre om den fått tränas på hela datan.

Det går även att diskutera såväl fördelar som nackdelar med att inte förbehandla datasetet mer grundligt, samt dess eventuella inverkan på resultatet. Eftersom datasetet var balanserat och både SVC och Random Forest modeller är effektiva på att hantera högdimensionella dataset, beslutades att endast skala datan och inte genomföra ytterligare behandling. Dimensionsreducering (PCA) övervägdes för att minimera bruset i datasetet (mycket bakgrund i bilderna) och endast behålla de egenskaper som står för 95% av den totala variansen. Dock visade modellerna bra prestanda även utan dimensionsreducering, och eftersom SVC inte är lika känslig för brus i data och presterade bättre på valideringsdata så ansågs det inte nödvändigt att tillämpa dimensionsreducering.

5.2 Frågeställning 2: Kan en modell tränad på MNIST dataset generalisera till ny data i en Streamlit-applikation?

Den andra delen i projektet handlade om att integrera modellen i en Streamlit-applikation och låta den göra prediktioner på ny data. Även om SVC-modellen presterade väl på ”ny data” från MNIST, visade den något sämre prestanda när den konfronterades med bilder från webbkameran i Streamlit-applikationen. Utmaningen här ligger i behandlingen av de bilder som modellen konfronteras med. Eftersom modellen är tränad på MNIST-data är det avgörande att de nya bilderna bearbetas för att likna träningsdatan så mycket som möjligt. Trots att detta utfördes kvarstod dock vissa svårigheter med att prediktera rätt på bilder tagna med webbkameran, troligtvis på grund av faktorer så som dålig belysning, vilket resulterar i ojämn skuggning i bilderna. Dessutom hade de uppladdade bilderna tagits med en mobilkamera, vilken genererar betydligt högre kvalitet och upplösning på bilderna i jämförelse med den integrerade webbkameran. För att öka modellens förutsättningar att göra korrekta prediktioner på bilder tagna med webbkameran, skulle man kunna experimentera med mer avancerade ”threshold” tekniker och bildbehandlingsmetoder i framtida projekt.

6 Slutsatser

Sammanfattningsvis lyckades projektet framgångsrikt med att implementera maskininlärning för att prediktera handskrivna siffror med hög noggrannhet. Detta både på MNIST-datasetet och genom att generalisera till ny data via en Streamlit-applikation. SVC-modellen visade sig vara mest lämplig för uppgiften och uppvisade höga generaliseringsförmågor vid utvärdering på testdata. Integreringen av modellen i en Streamlit-applikation visade hög potential, även om vissa förbättringar kan göras för att optimera modellens prestanda på bilder tagna via webbkamera.

Under projektets gång har jag bland annat lärt mig vikten av en utförlig EDA för att få en djupare förståelse för datan. Insikterna från den initiala dataanalysen är något som påverkar metodvalen i projektet, så som val av modell, utvärderingsmått och så småningom även behandlingen av ny data (nya bilder) som modellen ska konfronteras med.

För framtida projekt rekommenderas att experimentera med mer avancerade bildbehandlingsmetoder och ”threshold” tekniker för att förbättra modellens förutsättningar och prestanda ytterligare. Detta skulle kunna bidra med en ökad noggrannhet och robusthet i modellens prediktioner, särskilt när den konfronteras med bilder av sämre kvalitet.

7 Teoretiska frågor

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

Träningsdata används för att träna modeller, valideringsdata för att välja ut den bäst lämpade modellen och testdata för att utvärdera den slutgiltiga modellens 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”?

Hon kan använda korsvalidering (”cross-validation”) för att utvärdera modellernas prestanda och välja den med bäst prestanda, sett utifrån ett fördefinierat utvärderingsmått.

3. Vad är ”regressionsproblem? Kan du ge några exempel på modeller 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 (Y) som en funktion av dess ålder (X).

4. Hur kan du tolka RMSE och vad används det till: $RMSE$ = √∑(yi − $\hat{y}i$) 2 i

RMSE (Root Mean Squared Error) mäter skillnaden mellan de av modellen predikterade värdena och de faktiska värdena i datasetet, och används som mått för att utvärdera prestandan på modeller inom regressionsproblem. Ju lägre RMSE-värde, desto mindre fel gör modellen.

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

I klassificeringsproblem har den beroende variabeln kategoriska (diskreta) värden och vi försöker prediktera en eller flera klasser. Modeller som kan användas är exempelvis ”Support Vector Machine” och ”Logistic Regression”. Exempel på ett problem kan vara att modellera en blommas klasstillhörighet som en funktion av dess bladlängd och bladbredd. En ”Confusion Matrix” används för att mäta modellens prestanda genom att visa antalet rätt och fel klassificeringar.

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

K-means används för att dela upp data i K antal kluster i unsupervised learning. Det kan till exempel användas för att gruppera kunder med liknande egenskaper, vilket sedan kan användas för att rikta marknadsföring till kunder med en specifik grupptillhörighet.

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

Samtliga är metoder som används för att hantera kategorisk data. Låt säga att vi har ett dataset med den kategoriska variabeln ”ocean proximity”. I ordinal encoding får respektive kategori en unik siffra (inland 1, near ocean 2, near bay 3). Med one-hot encoding transformeras kategorierna till att representera varsin binär kolumn (inland [1,0,0], near ocean [0,1,0], near bay [0,0,1]). Dummy variable encoding består likt one-hot encoding av binära variabler men används genom att

droppa en kategori som sedan fungerar som en referenskategori (inland [1,0], near ocean [0,1]), dvs om båda variabler är 0 så betyder det ”near bay”.

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. Ett annat exempel är djur (ko, gris, hund, katt) som generellt sett inte har någon inbördes ordning (nominal data), men om jag exempelvis ber någon rangordna vilket djur de tycker är bäst så är datan ordinal, för då finns den en viss ordning. Det beror alltså på kontexten.

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 plattform som kan användas för att skapa webbapplikationer för projekt inom Data Science och maskininlärning.

8 Självutvärdering

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

Den största utmaningen under arbetets gång var att lista ut hur man skulle transformera de nya bilderna som modellen skulle prediktera. Jag hanterade det genom att backa tillbaka och utföra en mer detaljerad EDA, för att få insikt i hur träningsdatan ser ut och hur den är lagrad. Då framkom bland annat att träningsdatan är inverterad, dvs lagrad som vita siffror mot en svart bakgrund, vilket jag hade missat i min primära analys. Sen var det mycket tid som gick åt att läsa på om olika bildbehandlingsmetoder och testa med olika ”threshold” funktioner, för att reducera ”brus” i bilderna. I slutändan så valde jag att använda en ganska simpel metod genom att använda medelvärdet på pixlarnas intensitet.

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

Jag anser att jag har uppfyllt kriterierna för VG.

3. Något du vill lyfta fram till Antonio?

Det har varit ett roligt (och utmanande) projekt att arbeta med. Jag har lärt mig massor!

Källförteckning

Alsaafin, A., & Elnagar, A. (2017). A Minimal Subset of Features Using Feature Selection for Handwritten Digit Recognition. *Journal of Intelligent Learning Systems and Applications*, 9(4), 1-11.

Demirkaya, K., & Çavuşoğlu, Ü. (2022). Handwritten Digit Recognition with Machine Learning Algorithms. *Academic Platform Journal of Engineering and Smart Systems*, 10(1), 9-18.

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

Yang, L. (2021). *The image classification of MNIST dataset by using machine learning techniques*. University of California.