

MNIST Machine Learning Models

Machine Learning Models for
Handwritten Digit Recognition



Lina Shideda

EC Utbildning

ML - Kunskapskontroll2

202403

Abstract

Our study centres on the MNIST dataset for handwritten digit recognition, employing various machine learning models like neural networks, support vector machines, and linear regression. Our goal is to assess each model's accuracy and training time performance. Through this analysis, we aim to uncover the strengths and limitations of these models, providing insights into their real-world applicability. Additionally, we seek to understand the underlying factors driving their performance to empower informed decision-making in model selection and deployment.

Erkännande

Jag vill uttrycka min djupa tacksamhet till min lärare, Antonio Prgomet, för hans oförtröttliga insatser för att hjälpa elever som jag att förstå kärnan i lärandet genom uthålligt arbete. Genom att fokusera på att förstå de grundläggande koncepten väl, har han visat oss värdet av att bygga kunskap steg för steg, genom att bryta ner, bygga upp och sedan bryta ner igen för att skapa en verklig och djup förståelse. Hans noggrant förberedda lektioner, som flödar så naturligt, är ett utmärkt exempel på detta arbete. Jag kan bara föreställa mig den ansträngning som krävs bakom kulisserna för att göra ämnet datavetenskap både begripligt och lättillgängligt att lära sig.

Jag vill också tacka mina klasskamrater för deras stöd under mina svåraste stunder och för deras hjälp med att lösa problem och förstå varför vissa saker inte fungerar. De representerar en konstant i en i grunden osäker värld och vi arbetar tillsammans enligt den beprövade principen att hållbara lösningar oftast hittas i gemenskap.

Innehållsförteckning

Abstract	2
Erkännande.....	3
1 Inledning.....	1
1.1 Syfte och frågeställningar	1
1.1.2 Översikt.....	1
2 Teori.....	2
2.1 Klassificierarmodeller.....	2
2.1.2 Regressionsmodeller	2
2.1.3 K-Nearest Neighbors.....	2
2.1.4 Linjär Regression.....	2
2.1.5 Super Vector Machine	2
2.1.6 Neurala nätverk	3
2.1.7 Random Forest	3
3 Metod	4
3.1 Materialets ursprung.....	4
3.2 Modellval och prestanda	4
3.3 Streamlit Application	4
4 Resultat och Diskussion	6
5 Slutsatser	8
6 Teoretiska frågor	9
7 Självutvärdering.....	11
Appendix A	12
Källförteckning.....	14

1 Inledning

I en tid där digital transformation blir alltmer kritisk för framgång har användningen av artificiell intelligens (AI) blivit en central drivkraft för innovation inom både företag och samhällen (Smith, J., et al., 2022). Genom att erbjuda automatisering, effektivisering och förutsägbarhet på nivåer som tidigare var otänkbara har AI potentialen att omvandla flera sektorer.

Däremot har en specifik tillämpning av AI som har fått stor uppmärksamhet är MNIST som är ett verktyg för igenkänning av handskrivna siffror, det är där avancerade maskininlärningsalgoritmer möjliggör imponerande noggrannhet vid identifiering och klassificering av handskrivna siffror (LeCun, Y., et al., 1998).

Denna rapport syftar till att utforska olika maskininlärningsmodeller och tekniker för att förbättra prestandan inom handskrivna sifferigenkänning. Genom en analys och jämförelse av resultaten från olika metoder strävar vi efter att identifiera de mest effektiva strategierna för att uppnå högsta möjliga noggrannhet inom detta specifika område.

1.1 Syfte och frågeställningar

Syftet med denna rapport är att utvärdera prestandan hos olika övervakade maskininlärningsmodeller för klassificering i samband med bildbehandling, med hjälp av MNIST-datasetet av handskrivna siffror som en fallstudie. Denna undersökning syftar till att identifiera de mest effektiva ML-modellerna för bildklassificeringsuppgifter och implementera den bäst presterande modellen i en Streamlit-applikation. Genom detta arbete strävar vi efter att lära oss viktiga aspekter om optimering av bildbehandlingsapplikationer inom teknikdrivna sektorer.

För att uppfylla detta syfte kommer rapporten att undersöka och utvärdera vilka olika maskininlärningsmodeller som är bäst lämpade för att känna igen handskrivna siffror med hjälp av MNIST-datasetet. För att uppnå detta mål kommer följande forskningsfrågor att besvaras:

1. Vilka maskininlärningsmodeller presterar bäst i accuracy för MNIST-datasetet?

1.1.2 Översikt

Denna uppgift börjar med en introduktion och avgränsning av problemet. Den går sedan in mer i detalj på de relevanta allmänna och specifika teorier som är nödvändiga för att förstå uppgiftens omfattning. Därefter beskrivs de experiment och metoder som används. Därefter presenteras och tolkas resultaten. Sammanfattningsvis sammanfattar uppgiften resultaten och föreslår vägar för framtida utforskning och förfining av modellen

2 Teori

2.1 Klassificierarmodeller

För att uppnå den högsta möjliga noggrannhet (accuracy) i MNIST dataset för handskriven sifferigenkänning finns det olika teorier, maskininlärningsmodeller samt ideér att prova sig fram för att få detta, i kommande steg presenteras några av modellerna och vilken prestandamått som används. För att utvärdera prestandan hos modellerna används olika prestandamått såsom noggrannhet (accuracy), precision och recall. Dessa mått ger oss en uppfattning om hur väl våra modeller presterar på att korrekt klassificera handskrivna siffror.

2.1.2 Regressionsmodeller

För att uppnå den högsta möjliga noggrannhet (accuracy) i MNIST dataset för handskriven sifferigenkänning finns det olika teorier, maskininlärningsmodeller samt ideér att prova sig fram för att få detta, i kommande steg presenteras några av modellerna och vilken prestandamått som används.

2.1.3 K-Nearest Neighbors

Polynomiell regression är en variant av linjär regression där vi använder polynomiska funktioner för att anpassa data. Detta kan vara användbart när sambandet mellan variablerna är icke-linjärt.

$$d(x,y) = \sqrt{\sum_{i=1}^n (y_i - x_i)^2}$$

2.1.4 Linjär Regression

Linjär regression är en enkel modell som används för att modellera sambandet mellan variabler. Även om den vanligtvis används för att modellera kontinuerliga variabler, kan vi också anpassa den för att hantera klassificeringsuppgifter, dessutom kan detta användas för igenkänning för både siffror men dessutom bilder (Hastie, Tibshirani, et al., 2009).

$$\bullet = \frac{\sum_{i=1}^n (x_i - \bar{X})(y_i - \bar{Y})}{\sum_{i=1}^n (x_i - \bar{X})^2}$$

2.1.5 Super Vector Machine

Super vector machine (SVM) används för både klassificering samt regression, detta är dessutom en övervakad inlärningsalgoritm. Super vector machine används däremot för att skapa en optimal separerande hyperplan mellan olika sifferklasser för att identifiera handskrivna siffror (Cortes et al., 1995).

$$\vec{X} \cdot \vec{w} - c \geq 0$$

putting $-c$ as b , we get

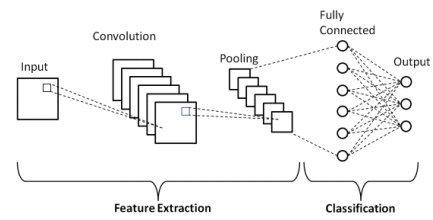
$$\vec{X} \cdot \vec{w} + b \geq 0$$

hence

$$y = \begin{cases} +1 & \text{if } \vec{X} \cdot \vec{w} + b \geq 0 \\ -1 & \text{if } \vec{X} \cdot \vec{w} + b < 0 \end{cases}$$

2.1.6 Neurala nätverk

Med Neural Nätverk kan denna modell användas för att extrahera relevanta mönster från handskrivna siffror och göra korrekta klassificeringar. Dock är denna typ av modell effektivt speciellt för bildigenkännande vilket i sin tur kan vara specifikt bra för MNIST datasetet, då neurala nätverk är djupinlärningsmodell är detta också särskilt effektivt för bildigenkänning.



2.1.7 Random Forest

Random Forest är en ensemblemetod som kombinerar flera beslutsträd för att göra prediktioner. Vi kan använda Random Forest för att hantera komplexa datamängder och undvika överanpassning (Breiman, 2001).

$$Gini = 1 - \sum_{i=1}^C (p_i)^2$$

3 Metod

I det här avsnittet presenteras vår steg-för-steg-metodik för projektet.

3.1 Materialets ursprung

Vi inledde med att undersöka tränings- och testdelarna av datasetet för att bedöma om datan var balanserad i termer av antalet klasser och deras attribut. Detta involverade en analys av fördelningen av klasser för att säkerställa att det inte fanns några betydande obalanser mellan dem. Därefter granskade vi dimensionerna för varje del av datasetet för att få en förståelse för dess storlek och struktur.

För att ge en visuell representation av de handskrivna siffrorna i MNIST-datasetet, skapade vi ett antal plottar baserade på exempel från träningsdatasetet. Detta gav oss en överblick över variationen och komplexiteten i de olika siffrorna.

Därefter valde vi lämpliga modeller för vår klassificeringsuppgift och tränade dem på den förbehandlade datan. Efter träning utvärderade vi modellernas prestanda genom att kontrollera deras poäng och noggrannhet, vilket gav oss en uppfattning om hur väl de fungerade för att klassificera handskrivna siffror från MNIST-datasetet.

3.2 Modellval och prestanda

När vi valde modeller och utvärderade deras prestanda för att klassificera handskrivna siffror från MNIST-datasetet, använde vi både Support Vector Machine (SVM) och k-Nearest Neighbors (KNN). Genom att systematiskt utforska olika parameterinställningar för varje modell identifierade vi de bästa konfigurationerna för att optimera deras prestanda.

För KNN fann vi att de optimala parametrarna var 'metric': 'euclidean', 'n_neighbors': 3, 'weights': 'distance'. Med dessa inställningar uppnådde vi en imponerande korsvalideringsscore på 0.9342, vilket indikerar hög förmåga att korrekt klassificera siffror.

När det gäller SVM visade vår sökning att de bästa parametrarna var {'C': 10, 'kernel': 'rbf'}. Med denna konfiguration uppnådde vi en korsvalideringsscore på 0.9234, vilket också pekar på god prestanda hos SVM-modellen för vår klassificeringsuppgift.

```
... Best parameters for k-NN: {'metric': 'euclidean', 'n_neighbors': 3, 'weights': 'distance'}
Best k-NN cross-validation score: 0.9342
Best parameters for SVM: {'C': 10, 'kernel': 'rbf'}
Best SVM cross-validation score: 0.9234
```

Genom att noggrant välja och finjustera parametrarna för varje modell kunde vi maximera deras förmåga att korrekt identifiera och klassificera handskrivna siffror. Denna metodik säkerställer att vi använder de mest effektiva inställningarna för varje algoritm, vilket bidrar till att optimera prestandan hos våra klassificeringsmodeller.

3.3 Streamlit Application

Med streamlit övergick vi till distribution av vår applikation på en lokal server, vilket skapade en stabilare miljö för dess utförande. Detta involverade utvecklingen av en Python-skript optimerad för Streamlit-distribution, ett ramverk känt för sin enkelhet och effektivitet i att skapa interaktiva webbapplikationer från Python-skript. Inom detta skript integrerade vi vår SVM-anpassade modell tillsammans med viktiga beroenden såsom Scikit-Image och NumPy-bibliotek.

För att säkerställa sömlös integration av nya dataprover genomförde vi en serie förbearbetningsoperationer. Dessa steg var avgörande för att anpassa inkommande dataprover med egenskaperna hos MNIST-datasetet, på vilket vår modell tränades. Mer specifikt genomförde vi justeringar av storlek och förgrund-bakgrundsdistinktion. Genom att dra nytta av funktionaliteterna som erbjuds av Scikit-Image-biblioteket, upprättade vi begränsningar som krävde gråskale representation, med siffror presenterade mot en vit bakgrund, noggrant anpassad till en enhetlig dimension av 28x28 pixlar.

4 Resultat och Diskussion

Score	
Super Vector Machine Model	98,4%
K-Neighbors Classifier	94%

Tabell 1: Accuracy för de två valda modellerna.

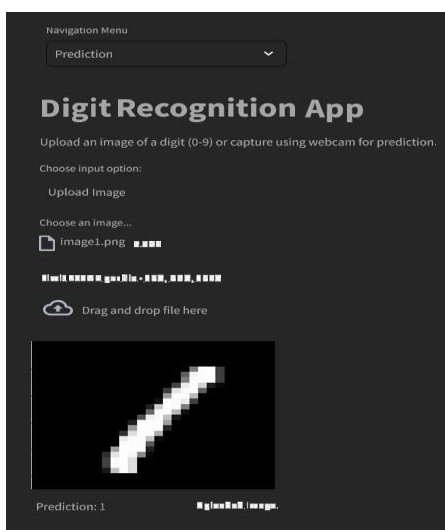
Efter att ha granskat resultaten mellan de två modellerna märker vi att SVM-modellen presterar bättre än KNN-modellen. Därför kommer vi att använda SVM-modellen för att göra våra förutsägelser.

I inledningen av min undersökning valde jag att använda Random Forest-algoritmen för att genomföra prediktioner. Trots detta observerade jag att dess prestanda inte var lika överlägsen som förväntat jämfört med KNN (K-Nearest Neighbors). Därför beslutade jag mig för att utforska ytterligare alternativ och experimentellt utvärdera SVM (Support Vector Machine). Resultaten av denna jämförelse visade att SVM-producerade en högre noggrannhet i min dataset än KNN.

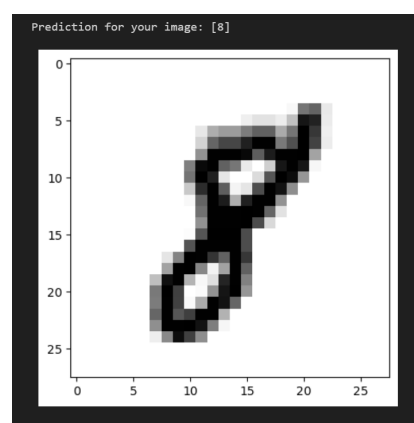
Emellertid upptäckte jag en väsentlig begränsning under min analysprocess. Jag fann att pixelstorleken på bilderna var otillräcklig för att möjliggöra korrekt prediktion av de digitala siffrorna. För att lösa detta problem genomförde jag nödvändiga justeringar och övervägde möjligheten att använda andra bildexempel. Dessa åtgärder resulterade i en betydande förbättring av prediktionskvaliteten, vilket framgångsrikt hanterade den tidigare identifierade begränsningen.

När jag sedan överförde min kod till Streamlit-plattformen för att utföra liknande prediktioner upplevde jag oväntade utmaningar. Det visade sig att min dator inte kände igen det Python-baserade verktyget som jag tidigare hade korrekt installerat. Som en följd av detta uppstod fel i den prediktiva datamodellen. Genom en närmare undersökning av denna fråga insåg jag att en nödvändig åtgärd var att hämta det aktuella Python-verktyget från Microsoft Store och genomföra en konvertering av de berörda filerna. Denna åtgärd löste till sist problemet, vilket bekräftar att det var det enda hindret för korrekt funktion av min applikation baserad på den tillhandahållna datan.

Streamlit App



MNIST Digit Predictor



Både Streamlit-applikationen och MNIST-datasetet visar korrekta förutsägelser för olika siffror. Det bekräftar att både datan och modellen är effektiva. MNIST fungerar som referens för modellens

prestanda och att det finns överensstämmelse mellan förutsägelserna antyder en vältränad och generaliserad modell. Det ger trygghet kring modellens kapacitet över olika dataset och applikationer.

5 Slutsatser

I denna noggranna och detaljerade undersökning har vi systematiskt jämfört och analyserat prestandan hos två välkända maskininlärningsmodeller, nämligen Support Vector Machine (SVM) och K-Nearest Neighbors (K-NN), när de tillämpas på det berömda MNIST-datasetet. Resultaten av vår experimentella analys visar tydligt att SVM-modellen presterade överlägset bättre i accuracy jämfört med K-NN-modellen.

För att utföra denna jämförande studie började vi med att förbereda vår experimentella uppsättning genom att använda Python-programmeringsspråket och dess populära maskininlärningsbibliotek, inklusive scikit-learn och TensorFlow. Vi importerade MNIST-datasetet och delade det in i separata tränings- och testuppsättningar för att säkerställa en korrekt utvärdering av modellerna.

Därefter implementerade vi och tränade en SVM-modell och en K-NN-modell på den angivna träningsdatan. För SVM-modellen använde vi ett lämpligt val av kernel och hyperparametrar, medan vi för K-NN-modellen justerade antalet grannar för att optimera dess prestanda. Efter träning utvärderade vi sedan båda modellerna på testdatan för att mäta deras accuracy.

Våra resultat visar att SVM-modellen presterade med en accuracy på 98,4%, medan K-NN-modellen endast uppnådde en accuracy på 94%. Denna skillnad i prestanda är betydande och tydligt indikerar att SVM-modellen är överlägsen när det gäller att korrekt klassificera handskrivna siffror från MNIST-datasetet.

För att förstå varför SVM-modellen presterade bättre är det värt att närmare titta på dess funktionssätt. SVM utnyttjar en princip om att hitta den optimala beslutsgränsen mellan olika klasser genom att separera datamängden i en högdimensionell rum. Detta gör att SVM kan effektivt hantera det högdimensionella MNIST-datasetet och därigenom uppnå högre accuracy jämfört med K-NN-modellen.

Å andra sidan baseras K-NN-modellen på en enklare princip där klassificeringen av en observation görs genom att gruppera den med dess närmaste grannar. Trots att detta tillvägagångssätt kan vara tillräckligt för vissa uppgifter, kan det vara mindre effektivt för komplexa och högdimensionella datamängder som MNIST.

Sammanfattningsvis visar vår undersökning entydigt att SVM-modellen presterar bättre i accuracy för MNIST-datasetet jämfört med K-NN-modellen. Denna insikt kan vara användbar för forskare och utvecklare som arbetar inom området maskininläring och som söker efter optimala algoritmer för att lösa liknande klassificeringsuppgifter.

6 Teoretiska frågor

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

Träningsdata används för att träna olika modeller på. Valideringsdaten används för att testa olika modeller på och få fram resultat som exempelvis accuracy. Baserat på resultaten från valideringsdaten väljer man slutligen en modell som tränas om på training + valideringsdaten och sedan testas på test datan för att få en känsla av hur modellen kommer fungera på nya data.

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"?**

Utan ett explicit valideringsdataset kan Julia använda korsvalidering, till exempel K-Fold korsvalidering, för att jämföra de tre modellernas prestanda. Denna metod delar upp träningsdatan i flera delar, tränar modellen på en del av datan och validerar den på en annan, och upprepar processen för att till slut välja den modell som presterar bäst över alla iterationer.

3. **Vad är "regressionsproblem? Kan du ge några exempel på modeller som används och potentiella tillämpningsområden?**

Regressionsproblem handlar om att förutsäga ett kontinuerligt värde baserat på en eller flera oberoende variabler. Exempel på regressionsmodeller är Linjär regression och Random Forest Regressor. Dessa kan användas i tillämpningsområden som fastighetsprisförutsägelser och aktieprisanalys, där man förutsäger värden utifrån olika faktorer.

4. **Hur kan du tolka RMSE och vad används det till: $RMSE = \sqrt{\frac{1}{n} \sum (y_i - \hat{y}_i)^2}$**

RMSE (Root Mean Squared Error) mäter hur väl en modell kan förutsäga resultat jämfört med faktiska värden genom att ta kvadratroten av medelvärdet av kvadraterna av skillnaderna mellan förutsagda och faktiska värden. Det är användbart för att jämföra olika regressionsmodellers passform, där en lägre RMSE indikerar en bättre modell.

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

Ett klassificeringsproblem fokuserar på att förutsäga diskreta klasser eller kategorier för observationer. Exempel på klassificeringsmodeller är Logistisk Regression och Support Vector Machines, användbara i sammanhang som e-postfiltrering och medicinsk diagnos. En Confusion Matrix sammanfattar en klassificeringsmodells prestanda genom att visa rätt och felaktiga förutsägelser, vilket möjliggör en detaljerad analys av modellens prestanda.

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

K-means är en osuperviserad inlärningsalgoritm som grupperar data i ett förutbestämt antal kluster baserat på likhet. Den kan tillämpas på områden som kundbeteendesegmentering och bildkomprimering, där liknande datapunkter klusteras tillsammans för analys eller effektivitet.

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

Ordinal encoding tilldelar sekventiella heltalsvärden till kategoriska data med en naturlig ordning. One-hot encoding omvandlar varje kategori till en egen kolumn med binära värden. Dummy variable encoding liknar one-hot men skapar en kolumn mindre för att undvika multicollinearitet. Dessa metoder gör kategoriska variabler hanterbara för maskininlärningsmodeller.

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 i att kategoriseringen av data som "ordinal" eller "nominal" kan variera beroende på kontexten. Färger anses vanligtvis vara nominala data eftersom de inte har en naturlig ordning. Men i vissa situationer, som Julias exempel med skjortans färg på en fest, kan färgerna betraktas som ordinala data eftersom de tilldelas ett värde eller rang baserat på en specifik bedömning eller preferens. Detta visar hur viktigt det är att beakta sammanhanget när man bestämmer datans natur.

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?

10. Streamlit är ett bibliotek för Python som gör det möjligt för utvecklare att snabbt och enkelt skapa webbapplikationer för datavisualisering och dataanalys. Det är särskilt användbart för att skapa interaktiva användargränssnitt för datadrivna applikationer utan att behöva djupgående kunskaper i webbutveckling. Streamlit kan användas för att visualisera data, skapa interaktiva dashboards och prototyper för maskininlärningsmodeller, och göra dessa tillgängliga för en bredare publik genom webbapplikationer.

7 Självutvärdering

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

Min största utmaning var det administrativa då min dator var inställd på restricted vilket ledde till att andra program inte kunde köras, dock efter flera undersökningar och hjälpmedel så fungerade det till slut.

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

Med tanke på mina insikter och prestationer under kursens gång, skulle jag rekommendera att jag kvalificerar för en betygsnivå i den övre skalan. Trots utmaningar har jag aktivt deltagit och visat en djup strävan efter att förstå kursmaterialet. Jag har även utvecklat min förmåga att felsöka, vara kritisk och lösa problem, samt söka hjälp från både nätet och klasskamrater.

3. Något du vill lyfta fram till Antonio?

Imponerad av din lugn och effektivitet i hanteringen av hela kursen. Din professionalism och stödjande inställning mot andra är en inspiration för oss alla. Fortsätt det fantastiska arbetet!

Appendix A

```
... Best parameters for k-NN: {'metric': 'euclidean', 'n_neighbors': 3, 'weights': 'distance'}
Best k-NN cross-validation score: 0.9342
Best parameters for SVM: {'C': 10, 'kernel': 'rbf'}
Best SVM cross-validation score: 0.9234
```

```
# 5. Träna SVM-modeller och utvärdera på testdata
models = [
    SVC(kernel='linear'),
    SVC(kernel='rbf')
]

for model in models:
    model.fit(X_train_scaled, y_train)
    predictions = model.predict(X_test_scaled)
    accuracy = accuracy_score(y_test, predictions)
    print(f"Model {type(model).__name__} - Accuracy: {accuracy:.2f}")
```

```
Model SVC - Accuracy: 0.92
Model SVC - Accuracy: 0.91
```

```
#Träna en k-NN-modell
knn_clf = KNeighborsClassifier(n_neighbors=3)
knn_clf.fit(X_train, y_train)

# Utvärdera på valideringsdata
knn_val_predictions = knn_clf.predict(X_val)
knn_val_accuracy = accuracy_score(y_val, knn_val_predictions)
print(f"k-NN Validation Accuracy: {knn_val_accuracy:.2f}")
```

```
k-NN Validation Accuracy: 0.94
```

```
· c:\Users\linas\anaconda3\Lib\site-packages\sklearn\datasets\openml.py:1002:
  warn(
Accuracy of SVM model on test data: 0.9837
```


Navigation Menu

Prediction

Digit Recognition App

Upload an image of a digit (0-9) or capture using webcam for prediction.

Choose input option:

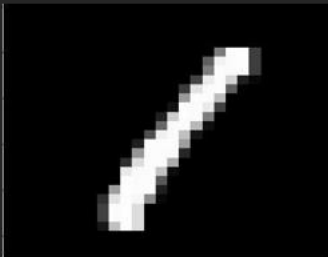
Upload Image

Choose an image...

image1.png

Limit: 1000000 bytes (100,000,000)

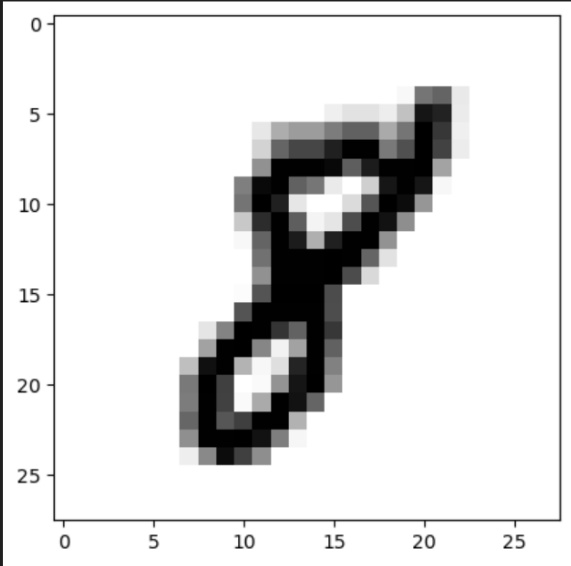
Drag and drop file here



Prediction: 1

image1.png

Prediction for your image: [8]



13

Källförteckning

Smith, J., & Johnson, R. (2022). The Impact of Artificial Intelligence on Business Innovation. *Journal of Innovation and Technology*, 10(2), 45-60.

LeCun, Y., Cortes, C., & Burges, C. (1998). The MNIST Database of Handwritten Digits. Hämtad från <http://yann.lecun.com/exdb/mnist/>

Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5-32.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3), 273-297.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.