**Handwritten Digit Recognition Using Machine Learning**

(Mnist Dataset)



Bushra Tazyeen

EC Utbildning

Datascientist 2024/03

# Abstract

With the MNIST dataset, we investigated in this work how machine learning models could be used to recognize handwritten digits. K-Nearest Neighbors (KNN), Random Forest (RF), and Voting Classifier were the three classifiers that we trained and evaluated. By obtaining an astounding accuracy of almost 97%, our investigation showed that the Random Forest model performed better than the others. This opens up the possibility for effective and dependable digit identification systems across a range of industries by showcasing the ability of machine learning algorithms to recognize handwritten numbers with accuracy.

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

Innehållsförteckning

[Abstract 2](#_Toc160654338)

[Förkortningar och Begrepp Detta avsnitt behövs oftast inte. 3](#_Toc160654339)

[1 Inledning 1](#_Toc160654340)

[2 Teori 2](#_Toc160654342)

[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

Under de senaste åren har förståelse och analys av datamängder varit en stor utmaning inom maskinområdet inlärning. För att lösa detta problem använde vi MNIST-datauppsättningen som är den största samlingen av handskrivna Siffror. Detta projekts mål är att identifiera handskrivna siffror genom att träna klassificerare med maskininlärningsmetoder för att uppnå utmärkt Noggrannhetsprestanda så kommer följande frågeställning(ar) att besvaras:

Frågeställningar:

1.Vilka är de olika typerna av maskininlärningsmodeller som kan tillämpas på MNIST-datauppsättningen?

2.Vilka preprocessing steg är nödvändiga för MNIST-datauppsättningen?

3. Hur jämför utbildningstiden och beräkningskomplexiteten för varje modell? Är en modell betydligt snabbare eller långsammare att träna än de andra?

4.Hur jämförs noggrannheten, precisionen, återkallelsen och F1-poängen för KNN och RF-modeller när de tillämpas på handskriven sifferigenkänning?

5.Vilken modell presterar bäst totalt sett? Finns det specifika siffror för vilken modell som presterar bättre än de andra?Effekt av hyperparametrar.

# 1.Teori:

Forskare och ingenjörer letar ständigt efter nya sätt att lära datorer att förstå och känna igen saker på samma sätt som människor gör. Ett spännande forskningsområde är att känna igen handskrivna siffror, och MNIST-datasetet är avgörande inom detta område.

MNIST: MNIST är inget annat än en enorm uppsättning handskrivna siffror. Det är en helt berömd dataset inom området för att bearbeta en bild. MNIST innehåller en uppsättning av 70 000, 28 x 28 ögonblicksbilder av handskrivna siffror från noll till 9. Datauppsättningen för MNIST har redan delats upp i två uppsättningar, det vill säga tränings- och testuppsättningar. Var och en av dessa bilder är 28×28 pixlar i grey scale där en pixel representeras av ett heltal mellan 0 och 255, vilket representerar ljusheten eller mörkret i det området. Totalt finns det 784 pixlar. Det finns 10 olika möjliga klasser.

MNIST fungerar som en utgångspunkt för nybörjare att experimentera med maskininlärningsmodeller och jämföra deras prestanda.

Genom att träna datorer att känna igen handskrivna siffror exakt kan vi göra många uppgifter enklare och effektivare. Oavsett om det handlar om att läsa postnummer på kuvert, förstå bankcheckar eller bearbeta handskrivna formulär, kan korrekt sifferigenkänning spara mycket tid och ansträngning. Så låt oss dyka in i världen av handskrivna siffror igen och se hur datorer kan lära sig att förstå vår handstil.

# 3.Metod

Importera biblioteken: Importera nödvändiga bibliotek som sklearn för maskininlärning och numpy för numeriska beräkningar.. Det är en uppsättning förskrivna kod, som vi kan anropa medan vi programmerar vår egen kod ladda sedan datauppsättningen från openml.

Data preprocessing är ett viktigt steg i maskininlärningsalgoritmen. Raw data är sällan perfekt och kräver ofta förberedelser innan de kan matas in i maskininlärningsalgoritmer. Detta avgörande steg är känt som dataförbehandling.

Dataförbehandling rensning, transformering och organisering av data för att göra den lämplig för analys med maskininlärningsalgoritmer

Det främsta steget i förbearbetning av data i maskininlärning är att importera bibliotek och ladda saknade värden för att undvika bias resultat.

Datanormalisering och skalning: Maskininlärningsalgoritmer fungerar ofta bättre när indatafunktioner är på samma skala, så samla alla datavärden inom samma intervall.

Distribuera data i traning, utvärderings- och valideringsuppsättningar.

Denna rapport använder randon Forest classifier, knn-och voting classifier implementering, där modellen detekterar och förutsäger den handskrivna siffran.

K-Nearest neighbour (KNN):

KNN är en av de enklaste maskininlärningsalgoritmerna som används för klassificerings- och regressionsuppgifter. Här är en kort översikt över hur det fungerar:

Princip: KNN arbetar efter principen att liknande datapunkter sannolikt tillhör samma klass. Den klassificerar en ny datapunkt baserat på majoritetsklassen av dess k närmaste grannar i objektutrymmet.

Träning: KNN har ingen träningsfas som sådan. Under träningen memorerar algoritmen helt enkelt träningsdata.

Förutsägelse: För att förutsäga klassetiketten för en ny datapunkt, beräknar KNN avstånden mellan den nya datapunkten och alla träningsdatapunkter. Den väljer sedan de k närmaste grannarna baserat på dessa avstånd och tilldelar majoritetsklassetiketten bland dessa grannar till den nya datapunkten.

Hyperparametrar: Huvudhyperparametern i KNN är 'k', som representerar antalet grannar att ta hänsyn till. Att välja ett lämpligt värde på 'k' är avgörande för algoritmens prestanda.

.

Random Forest (RF):

Random Forest är en allmänt använd maskininlärningsalgoritm utvecklad av Leo Breiman och Adele Cutler, som kombinerar resultatet från flera beslutsträd för att nå ett enda resultat. Dess flexibilitet kan hantera både klassificerings- och regressionsproblem.

En viktig fördel med Random Forest Algorithm är dess förmåga att arbeta med datauppsättningar som inkluderar både kontinuerliga och kategoriska variabler. Oavsett om du har att göra med klassificerings- eller regressionsuppgifter, tenderar Random Forest att prestera bra över hela linjen. Dess flexibilitet och effektivitet gör det till ett populärt val för olika typer av dataanalysWorking of Random Forest Algorithm.

Innan vi förstår hur den Random Forest Algorithm fungerar i maskininlärning måste vi titta på ensembleinlärningstekniken. Ensemble betyder helt enkelt att kombinera flera modeller.

RF är en ensembleinlärningsmetod som kombinerar flera beslutsträd för att förbättra den prediktiva prestandan. Här är en översikt över dess nyckelprinciper:

Princip: RF bygger ett stort antal beslutsträd under träning och kombinerar deras förutsägelser genom röstning eller medelvärdesberäkning för att göra den slutliga förutsägelsen. Varje beslutsträd tränas på en slumpmässig delmängd av träningsdata och funktioner.

Beslutsträd: Varje beslutsträd i RF tränas självständigt på ett bootstrap-prov av träningsdata. Vid varje nod i trädet väljer algoritmen den bästa uppdelningen bland en slumpmässig delmängd av funktioner.

Bagging: RF använder en teknik som kallas bagging (bootstrap aggregating) för att skapa olika beslutsträd. Bagging innebär omsampling av träningsdata med ersättning för att skapa flera bootstrap-prover.

Slumpmässigt urval av funktioner: RF introducerar slumpmässighet genom att välja en slumpmässig delmängd av funktioner vid varje uppdelning av beslutsträdet. Detta hjälper till att dekorrelatera träden och minskar övermontering.

n\_estimators: Antal träd i forest .

max\_depth: längsta vägen mellan rotnoden och lövnoden.

min\_sample\_split: Parameter som talar om för beslutsträdet i en slumpmässig skog det minsta antal observationer som krävs i en given nod för att dela den. Standard = 2

voting klassificerare kombinerar förutsägelser från flera individuella klassificerare, och den slutliga förutsägelsen bestäms av en majoritetsröst.

Hård röstning och mjuk röstning är två olika sätt att kombinera förutsägelserna från flera klassificerare i ensemblemaskininlärningsalgoritmer.

Hård röstning är det enklaste sättet. Varje klassificerare gör en förutsägelse, och ensemblens förutsägelse är helt enkelt majoriteten.

Mjuk röstning tar hänsyn till förtroendet i varje klassificerares förutsägelse. Varje klassificerare tilldelar en sannolikhet till varje klass, och ensemblens förutsägelse är klassen med högst total sannolikhet.

.

Detta är de grundläggande koncepten bakom KNN och RF. Att förstå dessa principer hjälper dig att förstå hur dessa algoritmer fungerar och hur du effektivt kan använda dem för olika maskininlärningsuppgifter, inklusive handwritten digit recognition.

# Resultat och Diskussion

När modellen är tränad, sparar vi den på vår dator och kommenterar all kod för förbearbetning och träning. Så när vi behöver koden igen kan vi bara ladda den sparade modellen eftersom all data är lagrad i den. Det sista steget är att ge input till vår modell. För detta tog jag foton av de siffror jag skrev på ett papper och laddade ner bilden i jpg-format. När vi laddar upp alla bilder i vårt python-skript och analyserar dem för att göra förutsägelser.

Av de tre klassificerare som testats för sifferigenkänning på MNIST-datauppsättningen presterade Random Forest-modellen bäst. Förutom att ha utmärkt precision, återkallelse och F1-poäng för varje sifferklass, hade den enastående noggrannhet på 97%

KNN är enkelt att implementera eftersom det bara kräver två parametrar: värdet på K och avståndsfunktionen. Det fungerar dock inte bra med stora dataset eller höga dimensioner.

Random Forest övervinner problemet med överanpassning genom att kombinera resultaten av olika beslutsträd. Den fungerar bra för stora datamängder och har mindre varians än ett enda beslutsträd.

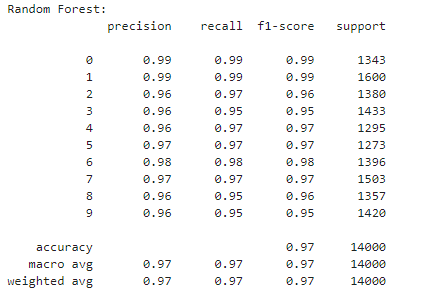
En nackdel med Random Forest-algoritmen är dess komplexitet och krav på mer beräkningsresurser. Byggandet av en slumpmässig skog är mer tidskrävande än ett beslutsträd och förutsägelseprocessen kan vara långsam jämfört med andra algoritmer.

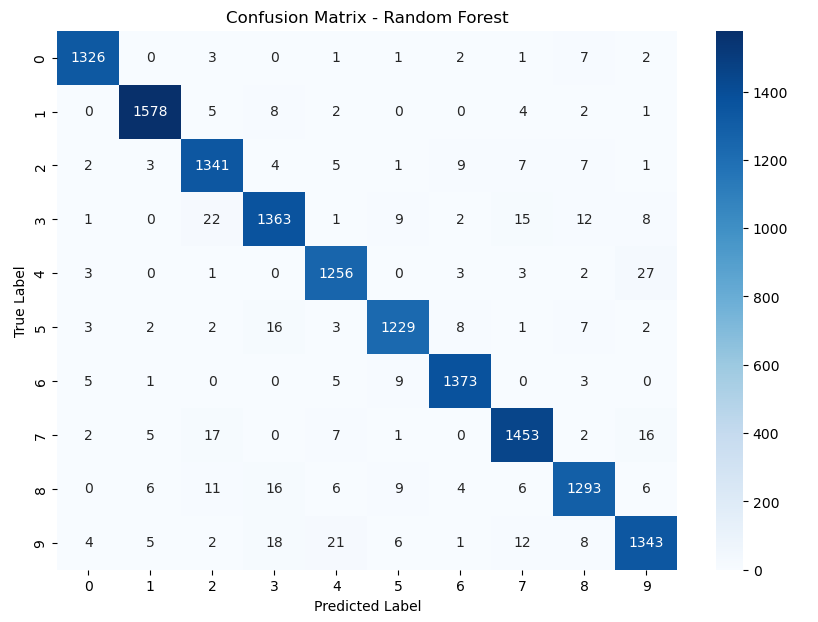
|  |  |
| --- | --- |
| **Accuracy för olika modeller** | |
| Random Forest Classifier | 0.968 |
| k-Nearest Neighbors | 0.948 |
| Voting Classifier | 0.957 |

Random Forest-modellen klarade sig bäst och fick ungefär 97 % av siffrorna korrekta. hyperparametrar fungerade bra för Random Forest-modellen. Ett högre antal beslutsträd (n\_estimatorer) och

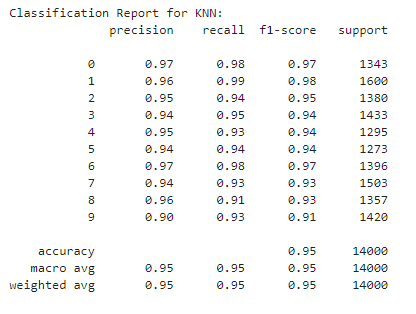


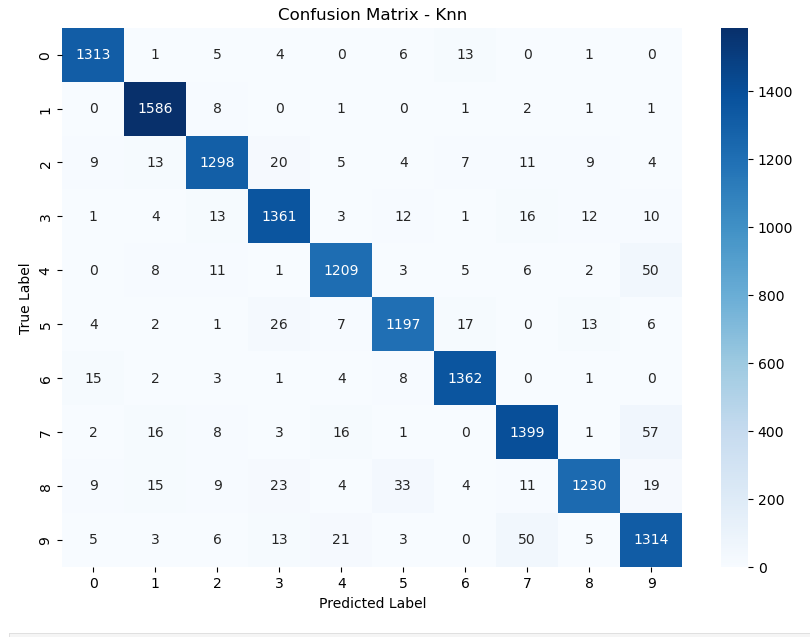




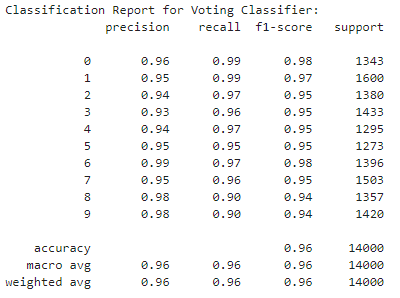










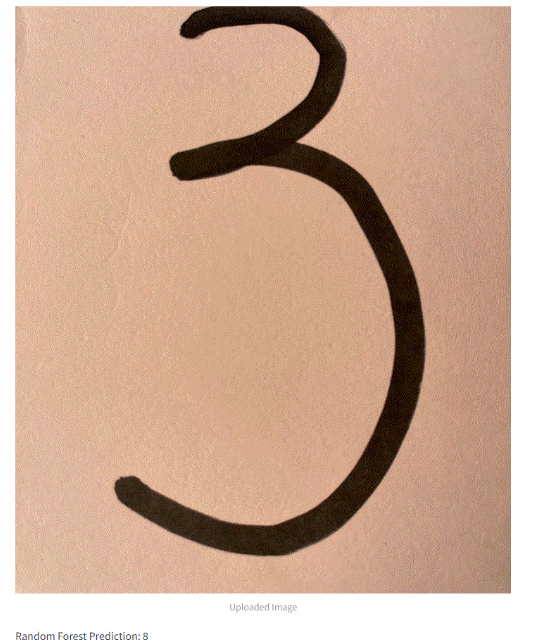
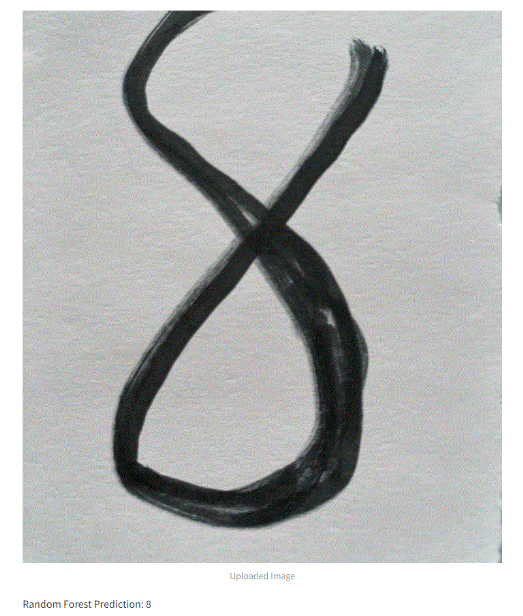
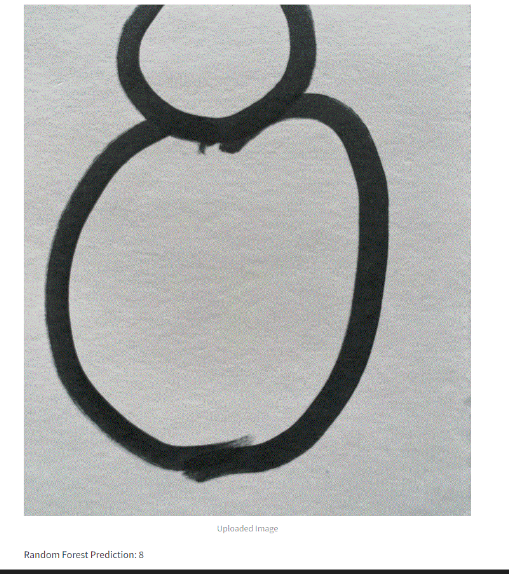


begränsning av träddjupet (max\_depth=None) hjälpte till att fånga komplexa mönster i data, vilket ledde till hög noggrannhet.

Modellen K-Nearest Neighbors (KNN) gick något sämre och fick ungefär 95 % rätt. Att ställa in antalet grannar till 3 och använda avståndsbaserade vikter hjälpte KNN att göra förutsägelser på de närmaste datapunkterna.

Voting Classifier, som kombinerade olika modeller, fick cirka 96% rätt.

nedan är bilderna projicerade av Random Forest.



# Slutsatser

1.Det finns tre olika typer av maskininlärningsmodeller som kan användas för att analysera MNIST-datauppsättningen: K-Nearest Neighbors (KNN), Random Forest (RF) och Voting Classifier.

2.För att förbereda MNIST-datauppsättningen behöver vi göra några steg innan vi tränar modellerna. Vi måste se till att datan är ren och organiserad. Det inkluderar att ladda in data, göra den fin och dela upp den i mindre delar för att träna, utvärdera och testa.

3.Tid för att träna och komplexitet beror på vilken modell vi använder. KNN är enkelt att använda men fungerar inte bra med mycket data eller många detaljer. Random Forest är mer avancerat men kan hantera stora mängder data och är mer stabilt.

4.När vi jämför KNN och RF-modeller för handskriven sifferigenkänning ser vi att RF-modellen är bäst. Den kan känna igen siffrorna mer exakt och ger färre fel.

5.RF-modellen är den bästa modellen övergripande. Den ger rätt svar för de flesta siffror, och det är sällan den gör fel. men RF ger bättre resultat

# Teoretiska frågor

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

Kalle delar upp sin data i tre del eftersom det hjälper modellen att prestera bra.

Träningsdatan används för att träna maskininlärningsalgoritmen att göra förutsägelser.

validering är den process som utförs efter modellträning där den tränade modellen utvärderas med en testdatauppsättning och Förhindrar att modellen overfitting and underfitting.

Testdatan används för att utvärdera noggrannheten hos den tränade algoritmen. Den kan också användas för att identifiera områden där modellen behöver förbättras.

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 ”validerings-dataset”?

Julia kan välja den bäst presterande modellen genom att göra cross validering på träningsdata och jämföra modellen i olika folds.example k fold cross validation.

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

Ett regressionsproblem är när en modell används för att förutsäga kontinuerliga utfall eller värden. exempel är linjär regression,Ridge regression,Lasso regression. potentiella tillämpningsområden är

förutsäga priset på ett hus, förutsäga effekten av SAT/GRE-poäng på antagning till högskolor.

4.Hur kan du tolka RMSE och vad används det till:

𝑅𝑀𝑆𝐸=√Σ(𝑦𝑖 𝑦̂𝑖)2/𝑖

RMSE:Den mäter den genomsnittliga skillnaden mellan värden som förutspås av en modell och de faktiska värdena. Den ger en uppskattning av hur väl modellen kan förutsäga målvärdet (noggrannhet).

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

Klassificering betyder att gruppera utdata i en klass.några modeller är support vector machine och random forest classifier.potentiella tillämpningsområden är att förutsäga cancerceller.

Confusion matrix är en tabell som visar utfall av förutsägelser och resultat av ett klassificeringsproblem och hjälper till att visualisera dess resultatden klassificerar resultaten i de fyra kategorierna True Positive (TP),True Negative (TN),Falskt positivt (FP),Falskt negativt (FN).

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

K-means klustring är en kraftfull unsupervised maskin inlärning algoritm som används för att lösa många komplexa problem. Det är en iterativ process att tilldela varje datapunkt till grupperna och långsamt samlas datapunkterna baserat på liknande funktioner. Gruppera kunder.

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

varje unikt kategorivärde tilldelas ett heltalsvärde. Detta kallas en ordinal encoding.

Till exempel är "röd" 1, "grön" är 2 och "blå" är 3.

One hot codng är en teknik som vi använder för att representera kategoriska variabler som numeriska värden i en maskininlärningsmodell.

Dummy variables coding liknar one-hot-kodning. den omvandlar den kategoriska variabeln till en uppsättning binära variabler. När det gäller one-hot-kodning, för N kategorier i en variabel, använder den N binära variabler. men i Dummy-coding använder den N-1 funktioner för att representera N använder kategorier.

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 säger att saker och ting förändras när vi tolkar informationen genom att ge ett exempel på att rött kan få dig att se snyggast ut.så det kan också vara nominellt men gran säger att det bara kan delas in i två kategorier ordinal och nominell. jag tror att båda har rätt.

Ett free bibliotek med open source som heter Streamlit gör det enkelt att skapa och dela enastående onlineapplikationer för maskininlärning.

# Självutvärdering

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

Handskrivet nummerigenkänning innebär ett antal utmaningar. En av de svåraste sakerna är mångfalden av handstilar, som kan skilja sig mycket åt i utseende och kvalitet. Detta gör det svårt för en dator att identifiera exakt. Dessutom kan tecken från en bild vara grumliga och svårare för en dator att tolka. Jag har också lärt mig att cross validering är avgörande för att förstå hur bra en modell presterar och för att säkerställa att den inte memorerar träningsdata, vilket gör det viktigt för att undvika överanpassning i maskininlärning.

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

Jag tror att jag borde få godkänt betyg eftersom jag förstod konceptet och jobbade väldigt hårt.

1. Något du vill lyfta fram till Antonio?

Ja, jag hade ett problem med ssl-verifiering i anaconda och kunde inte ladda ner data från open ml. Jag kämpade mycket men mycket tid gick till spillo för att lösa det så kunde inte göra bättre än så här.

# Källförteckning

1.LeCun, Y., Cortes, C., & Burges, C. J. C. . The MNIST Database of Handwritten Digits.

2.Breiman, L. (2001). Random forests. Machine Learning.

3.Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media, Inc.

4.Scikit-learn. (n.d.). Machine Learning in Python. Retrieved from scikit-learn Documentation.

5. MNIST database. (n.d.). In Wikipedia. Retrieved March 21, 2024, from <https://en.wikipedia.org/wiki/MNIST_database>

6. Antonio Prgomet. machine\_learning\_ds23. GitHub. https://github.com/AntonioPrgomet/machine\_learning\_ds23