Klassificeringsmodeller

En prestandaanalys på   
MNIST datasetet



EC Utbildning

Sofia Alves

Machine Learning Kunskapskontroll 2

20250321

# Abstract

This report will show how to solve a classification problem with the MNIST dataset. The models that are trained are Random Forest and Support Vector Machine in a Jupyter notebook using Python. In order to achieve a good score, the models have been configured with relevant parameters and hyperparameters. The best model is Random Forest, but Support vector machine’s accuracy score is not far behind.

Innehållsförteckning

[Abstract 2](#_Toc193452957)

[1 Inledning 1](#_Toc193452958)

[1.1 Bakgrund 1](#_Toc193452959)

[1.2 Om ämnet maskininlärning 1](#_Toc193452960)

[1.3 Varför är ämnet relevant? 1](#_Toc193452961)

[1.4 Syfte 1](#_Toc193452962)

[2 Teori 2](#_Toc193452963)

[2.1 Dataset 2](#_Toc193452964)

[2.2 Utvärderingsverktyg 2](#_Toc193452965)

[2.3 Hyperparametrar 3](#_Toc193452966)

[2.4 Random forest 3](#_Toc193452967)

[2.5 Support vector machine (SVM) 3](#_Toc193452968)

[3 Metod 4](#_Toc193452969)

[3.1 Modell nummer 1: Random forest 4](#_Toc193452970)

[3.1.1 Dataanalys 4](#_Toc193452971)

[3.1.2 Träna modellen 4](#_Toc193452972)

[3.1.3 Prediktera på testdata och kontrollera resultat 5](#_Toc193452973)

[3.2 Modell nummer 2: SVM 5](#_Toc193452974)

[3.2.1 Indelning av träningsdata 5](#_Toc193452975)

[3.2.2 Dataanalys 5](#_Toc193452976)

[3.2.3 Optimering och träning av modellen 6](#_Toc193452977)

[3.2.4 Prediktera på testdata och kontrollera resultat 6](#_Toc193452978)

[4 Resultat och Diskussion 7](#_Toc193452979)

[4.1 Random forest slutprestation 7](#_Toc193452980)

[4.2 Support vector machines slutresultat 7](#_Toc193452981)

[5 Slutsatser 8](#_Toc193452982)

[5.1 Funna resultat på Random forest och MNIST-datasetet: 8](#_Toc193452983)

[5.2 Funna resultat på SVM och MNIST-datasetet: 8](#_Toc193452984)

[5.3 Vilken modell hade högst score? 8](#_Toc193452985)

[5.4 Vilken modell hade flest problem med att identifiera bilderna i MNIST? 9](#_Toc193452986)

[5.5 Är en jämförelse möjlig att göra på de valda modellerna? 9](#_Toc193452987)

[5.6 En tankeställare 9](#_Toc193452988)

[6 Teoretiska frågor 10](#_Toc193452989)

[7 Självutvärdering 12](#_Toc193452990)

[7.1 Utmaningar du haft under arbetet samt hur du hanterat dem. 12](#_Toc193452991)

[7.2 Vilket betyg du anser att du skall ha och varför. 12](#_Toc193452992)

[7.3 Något du vill lyfta fram till Antonio? 12](#_Toc193452993)

[Appendix A: bilder och diagram. 13](#_Toc193452994)

[Källförteckning 19](#_Toc193452995)

# Inledning

## Bakgrund

Denna rapport är ett arbete i kursen om grundläggande maskininlärning som hålls av EC utbildning. En kurs som har varit utmanande men otroligt lärorik. Arbetet som har lagts ner i denna rapport har stärkt undervisningens teoretiska del, tvingat fram nya perspektiv och slutligen glädjen att förstå och analysera de valda modellernas prestation. Denna rapport kommer att redogöra hur två modeller (Random forest och Support vector machines) har tränats, testats och utvärderats.

## Om ämnet maskininlärning

Maskininlärning (Machine learning) är ett område inom AI och är en metod där maskiner lär sig från data. Det finns flera formella definitioner dock fångar följande förklaring på ämnet, på ett mycket bra sätt:  
"*Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention*."   
Statistical Analysis System (2025), Machine learning what it is and why it matters. Hämtad 13 mars 2025 från SAS sida <https://www.sas.com/en_us/insights/analytics/machine-learning.html>

AI började redan på 1950 då Alan Turing (brittisk matematiker) arbetade fram en idé (Turingtestet) om att maskiner kan ha ett intelligent beteende. Det var dock 1959 då Arthur Samuel (Nordamerikansk pionjär inom maskininlärning) myntade begreppet ”machine learning”. Utvecklingen har bara fortsatt med flera framsteg såsom chatbots, neurala nätverk, robotik och till dagens AI chatbotar såsom ChatGPT och Gemini.  
IBM, The history of AI, (2024). Hämtad 19 mars 2025 från <https://www.ibm.com/think/topics/history-of-artificial-intelligence>

## Varför är ämnet relevant?

Maskininlärningens framsteg i dagens samhälle har varit markant. En utveckling som kommer att fortsätta öppna nya dörrar och underlätta och påverka det dagliga livet i vårt samhälle. Flera funktioner som används dagligen är utvecklade med hjälp av maskininlärning. På denna lista finns bland andra mejlfiltrering, rekommendationer i onlinebutiker och bildigenkänning i vården.

## Syfte

Denna rapport kommer att visa hur de valda modellerna har kunnat identifiera bilderna i MNIST datasetet, som består av bilder av handskrivna siffror. Ett av målen är att undersöka hur en modell som inte är ett förstahandsval för bildklassificering presterar. För att uppfylla syftet kommer följande frågeställningar att besvaras:

1. Vilken modell har den bästa accuracy score?
2. Vilken modell hade flest problem med att identifiera bilderna i MNIST?
3. Är en jämförelse möjlig att göra på de valda modellerna?

# Teori

Maskininlärning brukar delas in i fyra kategorier: Supervised (med etiketterad data), Unsupervised (med oetiketterad data), Reinforcement (förstärkningsinlärning) och Semi-supervised learning (delvis övervakad inlärning) som är en kombination av både Supervised och Unsupervised learning.   
(Géron Aurélien, Hands on machine learning with Scikit learn, Keras and Tensorflow, 2:a upplagan, O’Reilly media, (sid 7-15) ), (Se bild nr 1).

## Dataset

Datan som har analyserats är MNIST- datasetet (bilder med handskrivna siffror, se bild nr 2) således valdes modeller som klarar av klassificeringsproblem. Dessa ingår I gruppen Supervised learning och de valda modellerna för denna rapport är Random forest och Support vector machine.

## Utvärderingsverktyg

För att utvärdera modellerna har confusion matrix, classification support och ROC (Receiver operating characteristic) kurvan använts. Confusion matrix visar antal korrekta och felaktiga prediktioner (se bild nr 3). Classification report visar en detaljerad bild på klasserna som undersöks samt vilka värden dessa har fått:

* Precision visar hur väl modellen predikterar verkliga prediktioner av alla positiva prediktioner.
* Recall visar faktiska positiva prediktioner av verkliga positiva prediktioner.
* F1 score visar balansen mellan precision och recall.
* Support visar klassernas fördelning.
* Macro avg är medelvärdet på klassernas prestanda.
* Weighted avg talar om balansen mellan klasserna.

ROC (Receiver operation characteristics) är en kurva som visar true positive rate och false positive rate. Den mäter modellens prestanda och här eftersträvas att få hög TPR (true positive rate) och låg FPR (false positive rate).

AUC (Area under the curve) mäter ytan under kurvan. Målet är att undersöka om modellen särskiljer klasserna väl och nå det optimala värdet 1.

## Hyperparametrar

För att optimera modellerna ytterligare används Random search på båda modellerna. En metod som testar slumpmässigt valda kombinationer och kan vara snabbare och effektivare än Grid Search. Hyperparametrarna som ställs in är följande:

* Cross validation (cv): delar in datan i träning och validering (folds)
* Verbose: antal meddelanden man får som output. kan ställas in som 0 (ingen output), 1 (minimal output) eller 2 (maximal output)
* N\_jobs: talar om hur många körningar som kan användas parallellt (-1 ingen begränsning, None används en)
* N\_iter: antal kombinationer som testas

## Random forest

En ensemblemetod som bygger på en samling av flera beslutsträd. Vid klassificeringsproblem använder man sig av en majoritetsröstning. Det innebär att varje träd tränas på ett urval av observationerna som delas in slumpmässigt in i flera beslutsträd (bootstrap sampling/återläggning). Därefter utförs en förgrening (split) som leder till beslutsunderlag. Formel för random forest classifier vid majoritetsröstning:

För att kontrollera en nods förmåga att separera klasserna används Gini parametern:

En annan parameter är entropy som mäter oordning i en nod:

## Support vector machine (SVM)

SVM tillhör gruppen supervised learning och den försöker hitta hyperplanet som separerar klasserna med maximala marginaler. SVC är en algoritm inom Support Vector Machine och möjliggör hantering av klassificeringsproblem med hjälp av kernel trick; RBF (som används i denna rapport), som mappar datan i en högre dimension.

Optimering i SVM (hitta den bästa linjen):

Under villkoret (punkterna hamnar på rätt sida av linjen):

RBF kernel hittar avståndet mellan två punkter och är kraftfullt vid komplexa icke-linjära problem:

# Metod

Målet är att undersöka valda modellers förmåga att identifiera handskrivna siffror i MNIST- datasetet. För att uppnå detta har jag tränat två modeller: Random Forest och Support Vector Machines. Modellerna är kodade i Jupyter notebook med Python och med importer från Scikit- learn.

Datan som har använts har laddats ner lokalt från Kaggle.com (av Joseph Redmon) och fördelad på två filer: mnist\_train (träningsdata) och mnist\_test (testdata) i csv-format. Träningssetet består av 60 000 bilder och testsetet av 10 000, där varje bild är 28 x 28 pixlar (totalt 784 per bild).

## Modell nummer 1: Random forest

Kodningen av random forest inleddes med att importera anpassade bibliotek för modellen. Laddning av datan i notebooken gjordes genom att dela in kolumnerna i x (features) och y (label) med en konvertering till Numpy- array (för att kunna hanteras av scikit- learn).

### Dataanalys

Dataanalysen består av att undersöka datatyper, antal värden, klassfördelning, etiketter och identifiera eventuella saknade värden.

### Träna modellen

Modellens parametrar justerades med följande:

* Antal beslutsträd sattes till 50, 100, 150, 200 (n\_estimators).
* Maximal djup för beslutsträden sattes till 10, 20, 30 nivåer (max\_depth ).
* Träden ställdes in att endast dela noder med minst 5 eller 10 datapunkter (min\_samples\_split).
* För att ett blad ska vara aktuell för användning ska det ha minst 2 eller 4 prover (min\_sample\_leafs).
* För att mäta hur noggranna noderna är (criterion) användes två kriterier:  
  1. Gini mäter graden av oordning i datan  
  2. Entropy informationsinnehållet i noderna.

För att optimera modellens prestanda användes Random search för att söka efter de bästa hyperparametrarna. Denna metod väljer slumpmässiga kombinationer av parametrar för att hitta den bästa konfigurationen.

* Cross validation med 10 folds
* Verbose= 1
* n\_jobs -1
* n\_inter 10

Resultatet från random search visade att de bästa parametrarna var:

* Criterion= ”gini”
* Max\_depth: 30
* Min\_samples\_leaf= 2
* Min\_samples\_split= 5
* N\_estimators= 100
* Random state= 42.

### Prediktera på testdata och kontrollera resultat

Nästa steg var att testa modellen på testdatan frånfilen mnist\_test och analysera hur väl modellen predikterar på ny data. För att kunna undersöka detta användes accuracy score, confusion matrix, classification report. (se bild nr 4 och 5)

För att ytterligare kontrollera modellens prestanda skapades en ROC- kurva (med one-versus-rest, varje klass mot alla andra) som visar förhållandet mellan True positive rate och false positive rate för varje klass. Det som eftersträvas är en kurva som är så nära 1 som möjligt (diagrammets översta vänstra hörn). (Se bild nr 6)

## Modell nummer 2: SVM

Nästa modell som tränas är Support Vector Machines. I vanlig ordning har specifika importer gjorts och träningsset och trainset har laddats in i X (features) och y (label).

### Indelning av träningsdata

Datan delades in i ett träningsset på 20 000 bilder. Testsetet är oförändrat. Sedan skapas en pipeline där standardscaler, PCA och RBF kärna har definierats därefter tränas detta minimerade set.

### Dataanalys

Eftersom MNIST-datasetet består av bilder har datan undersökts genom att ta fram klassfördelning, saknade värden och normalisering. Dessa steg är väldigt viktiga innan prediktion för att åtgärda eventuella punkter som kan påverka utfallet negativt. För att kontrollera att alla features och labels fungerar korrekt skapades en kod som verifierar detta.

### Optimering och träning av modellen

För att få ett ett balanserat resultat, har SVM parameter ställts in. Felaktiga värden kan göra kodkörningen långsam och trög. Dessa inställningar består av n\_components, c (regularization), kernel och gamma. Parametrarna blev till slut enligt följande:

* N\_components= 30,50,100 (här testar modellen olika komponenter som anges)
* C= 0.1, 1, 10 (hitta bästa marginalstorlek)
* SVM Kernel= RBF (sattes efter tester med både linear och RBF)
* SVM Gamma= sattes till 0.001, 0.01, 0.1. (styr hur träningspunkterna påverkar utfallet i RBF)

Hyperparametrar har ställts in i Random search för att förbättra träningen ytterligare. Resultatet blev: 'svm\_\_kernel': 'rbf', 'svm\_\_gamma': 0.001, 'svm\_\_C': 10, 'pca\_\_n\_components': 100

### Prediktera på testdata och kontrollera resultat

För att prediktera testdatan användes random searchs resultat med de bästa parametrarna. Utfallet av modellens prestation har sedan analyserats med hjälp av utvärderingsverktyg såsom confusion matrix, classification report (se bilder 7 och 8), plottdiagram (PCA med två dimensioner, diagram nr 1) och slutligen en tabell som visar alla värden för bästa möjliga parameterkombination som random search fick fram.

# Resultat och Diskussion

Tabell 1: Resultat av modellernas prestation.

Som tabell nr 1 visar, presterade Random forest bättre än Support vector machines med 0,18 procentenheter.

## Random forest slutprestation

Random forest fick en accuracy score på 96,63% och Confusion matrix hade en fin diagonal linje. I syfte att undersöka mer om felprediktionerna, togs det ut de 10 vanligaste. Modellen presterade sämre på siffrorna 3,4,5,7,8 och 9. Vidare visade Classification report en god balans mellan klasserna, Precision på 97%, Recall på 97% och F1 score 97%. Vidare kan man konstatera att även inställningarna i Random search bidrog till modellens slutresultat. De bästa inställningarna blev till slut rbf kernel, gamma: 0.01, C: 10, n\_components= 100.

Analysen av entropy (oordning i noderna) visade värden mellan 0,00 och 0,89. Dessa värden tydde på god ordning i noderna.

ROC kurvan visar att modellen hade hög True positive rate. I samma diagram kan man avläsa att Area under the curve (AUC) har värdet 1 genom alla klasser.

## Support vector machines slutresultat

Support vector machine nådde en score på 96,45% efter en del körningar. Innan optimering av diverse parametrar hade modellen 93,2%. Det som avgjorde denna förbättring var:

* Skalning av datan som undviker snedvridning av klasser, styr optimering mellan datapunkter, ger en snabbare körning och underlättar för RBF kerneln.
* PCA (principle component analys): reducerar dimensionerna och underlättar SVM:s prestation.

Confusion matrix visade att modellen är bra på att prediktera med en utmärkt diagonal linje. Denna modell hade flest felprediktioner på bilder med siffrorna 2,4,5,7,8 och 9 . Classification report visar att Precision på 96%, Recall på 96% och en F1 score på 96%. Ett plottdiagram skapades med hjälp av PCA för att undersöka klassernas fördelning vilket visade sig ha bra avgränsningar. (SE BILD)

# Slutsatser

Resultaten av modellerna prestation visade vara mycket bra. I syfte att kunna utvärdera modellernas score i denna rapport, gjordes en undersökning på internet.

## Funna resultat på Random forest och MNIST-datasetet:

Kaggle fick en score på 96,357% med liknande inställningar.  
Kaggle, (2018), MNIST Classification using Random Forest, hämtad 16 mars 2025 från <https://www.kaggle.com/code/ashwani07/mnist-classification-using-random-forest>

Medium fick en score på 96,7%, tyvärr framgår det inte vilka inställningar som har använts.  
Medium, (2023), Trying Out Both Random Forest & Decision Tree Classifier On MNIST Using Scikit-Learn, hämtad 16 mars 2025 från   
<https://devkyle4.medium.com/trying-out-both-random-forest-decision-tree-classifier-on-mnist-using-scikit-learn-21c8558c22e6>

Escholarship fick en score på 97,58% med liknande inställningar.   
Escholarship, (2021), The image classification of MNIST dataset by using machine learning techniques, hämtad 17 mars 2025 från <https://escholarship.org/content/qt0505d532/qt0505d532_noSplash_f17a9f77bcfc62b59306e4196c2cd6dc.pdf?t=r464cp>

## Funna resultat på SVM och MNIST-datasetet:

Dhairya Kothari i Github visar en score på 96,4% med liknande RBF inställningar.   
Github, (u.å), SVM with MNIST, hämtad 19 mars 2025 från  
<https://dmkothari.github.io/Machine-Learning-Projects/SVM_with_MNIST.html>

Nishan Patel på Kaggle.com fick en score på 94,38% med RBF kernel.  
Kaggle, (2019), MNIST Digit recognition using SVM, hämtad 19 mars 2025 från  
<https://www.kaggle.com/code/nishan192/mnist-digit-recognition-using-svm>

Sidhard på PyCodeMates fick en score på 94% med RBF kernel.  
PyCodeMates, (2022), The RBF kernel in SVM: A Complete Guide, hämtad 19 mars 2025 från  
<https://www.pycodemates.com/2022/10/the-rbf-kernel-in-svm-complete-guide.html>

Random forest resultat i denna rapport ligger i linje med liknande tester som hittades. Resultaten där SVM modellen har använts, är något lägre. Slutsatsen är att modeller har en tendens att prestera bra med MNIST- datasetet.

## Vilken modell hade högst score?

Random forest presterade bäst. Trots att SVM:s är inte ett förstahandsval när det kommer till större dataset och att klassificera bilder, fick modellen en hyfsat bra score med hjälp av inställningarna på parametrarna. SVM presterade sämre med linear och RBF kernel med hjälp av PCA lyfte upp prestandan.

## Vilken modell hade flest problem med att identifiera bilderna i MNIST?

Trots att båda modellerna hade svårigheter med nästan samma siffror var det ändå SVM som visade sig ha övervägande problem med identifieringen. (Se diagram 2 och 3)

## Är en jämförelse möjlig att göra på de valda modellerna?

Frågeställningen om man kan göra en rättvis jämförelse mellan dessa är helt klart ja. Trots att modellerna är olika och angriper problem på olika sätt (detta fall klassificeringsproblem) är min slutsats att det ligger mycket i hur väl man lyckas utnyttja varje modells styrkor och svagheter för att uppnå ett bra resultat.

## En tankeställare

Trots dessa goda resultat, som rapporten har visat (och ovannämnda testresultat på internet) vill jag ändå lyfta fram AUC värdet i Random forest modellen. Att samtliga klasser har värdet 1 kan verka för bra för att vara sant. I koden gjordes flera inställningar och kontroller för att undersöka överanpassning: indelningar av träningsdata och testdata, trädens parametrar och hyperparametrarna. Entropyvärdena (oordning i noderna), hade olika värden för varje gång parametrarna ändrades och med de slutgiltiga inställningarna kunde man se att noderna hade en bra separationsförmåga. Just denna relation mellan detta AUC värde och Entropyvärden är värt att undersöka vidare på.

# Teoretiska frågor

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

När modeller ska tränas och analyseras är indelningen bland det första man gör. Man tränar modellen med träningsdel, valideringsdelen används för att validera träningsdatan och slutligen använder man testdelen för att kunna analysera hur bra modellen predikterar på ny 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 ”validerings-dataset”?**

Oavsett vilken modell hon väljer och utan ett valideringsset behöver hon använda Cross validation. Den delar in datan i x antal folds (delar) med x antal iterationer och returnerar en score som talar om hur bra performance modellen har på ny data.

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

Ett regressionsproblem är när det finns ett förhållande mellan olika variabler och har en kontinuerlig output. Ex man utgår från Y (label) där x antal variabler (features) påverkar utfallet. Modeller som är anpassade för dessa sorters problem är bland andra:

* Linjär regression
* Logistic regression
* Lasso regression
* Ridge regression.

Tillämpningsområden: finansiella marknaden (aktier, huspriser), företag som vill exempelvis undersöka försäljningar, löner och effekterna av marknadsföring på intäkterna.

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

Roten ur summan av faktiska värden – predikterade värden upphöjt till 2 genom n observationer.

Root mean square error är ett mått på skillnaden mellan faktiskta värden mot predikterade värden. Används för att jämföra prestationer mellan modeller. Låg RMSE indikerar att modellen gör bra prediktioner.

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

Ett klassificieringsproblem är när man vill kolla data som delas in i två eller flera klasser (multiklassificering). Exempelvis att vilja kontrollera om ett objekt är blått eller gult. Tillämpningsområden: ansiktsigenkänning, mejlfiltrering, banker som vill inspektera om kunders transaktioner är ovanliga.

En confusion matrix är visar hur bra en modell predikterar mot faktiskta värden, genom att dela in resultaten i True positive, False negative, False positive och True negative.

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

Kmeans är en klustringsmetod som den delar in datapunkter i olika ”kluster” baserat på punkternas avstånd till centroiden. Kan används för att dela in kunder i olika grupper.

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

Ordinal coding: Omvandlar kategorisk data till numeriska värden och behåller deras ordning. Exempelvis betygssystem.

One hot encoding: Omvandlar kategorisk data till binär data med värdena 0 eller 1, utan ordning. Exempel att tala om en kolumn är röd, blå eller vit.

Dummy variable encoding: påminner om one hot encoding men har en kolumn mindre och talar om om en kategori är närvarande eller frånvarande.

**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?**

Görans påstående är inte helt korrekt då data kan anta flera former bland annat ordinal eller nominal. I Julias fall så stämmer det, tre kategorier utan inbördes ordning är nominal (utan ordning) men om vill värdera kategorierna (ex. röd = vacker), förvandlas dessa till ordinal.

**9.Kolla följande video om Streamlit: https://www.youtube.com/watch?v=ggDa-RzPP7A&list=PLgzaMbMPEHEx9Als3F3sKKXexWnyEKH45&index=12 Och besvara följande fråga: - Vad är Streamlit för något och vad kan det användas till?**

Streamlit är ett open- source verktyg där statistiker, maskinlärningsspecialister och data scientists kan skapa webbapplikationer i Python.

# Självutvärdering

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

De flesta utmaningarna var att hantera problem med Jupyter notebooken och koder som inte fungerade som jag ville. Jupyter hängde sig flera gånger, fick skapa ny miljö och starta om Anaconda flera gånger, vilket var mycket frustrande. Detta blev en tidstjuv, men det gick att lösa till slut. En annan utmaning var att brottas med mig själv! Jag började denna uppgift med att testa fem olika modeller av ren nyfikenhet. Förutom de valda modellerna testad jag även logistic regression, voting classifier och KNN. Att testa fem modeller och samtidigt ha tekniska problem blev väldigt jobbigt och ett stort stressmoment.

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

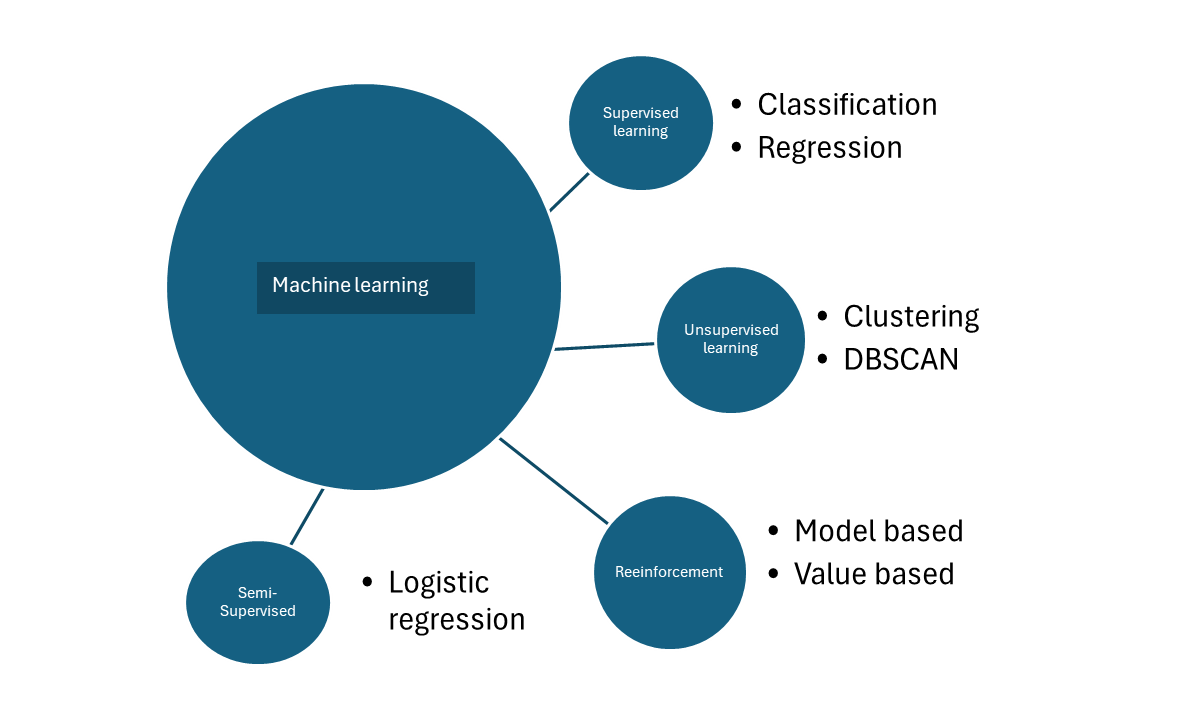
Jag hoppas på att få godkänd. Mitt mål är att förstå grunderna för varje kurs och min förhoppning är att denna rapport har cementerat detta. Genom att aktivt sitta och koda och se vad som händer är ju en helt fantastiskt för att förstå begreppen i maskininlärning.

## Något du vill lyfta fram till Antonio?

Tycker nog att allt var bra med kursen. Litteraturen var tung men när jag inte förstår ett begrepp så söker jag på internet för att få andra förklaringar. Alla videos på Youtube är jättebra gjorda och ett fantastiskt verktyg under studierna.

Vill även passa på och tacka för stödet ni har gett mig. Det är guld värt!

# Appendix A: bilder och diagram.



*Bild nr 1: Maskininlärningstyper*

  
*Bild nr 2: MNIST- datasetet. Bild hämtad* [*https://en.wikipedia.org/wiki/MNIST\_database*](https://en.wikipedia.org/wiki/MNIST_database)

*En bild som visar text, skärmbild, Teckensnitt, nummer

AI-genererat innehåll kan vara felaktigt.  
Bild nr 3: Confusion matrix.*

*En bild som visar text, skärmbild, Teckensnitt, nummer

AI-genererat innehåll kan vara felaktigt.  
Bild nr 4: Classification report.*

*En bild som visar skärmbild, kvadrat, diagram, Rektangel

AI-genererat innehåll kan vara felaktigt.  
Bild nr 5: Confusion matrix för Random Forest.*

*En bild som visar text, skärmbild, nummer, Teckensnitt

AI-genererat innehåll kan vara felaktigt.  
Bild nr 6: Classification report för Random Forest.*

En bild som visar text, skärmbild, Graf, linje

AI-genererat innehåll kan vara felaktigt.  
*Diagram nr 1: ROC kurva för Random Forest.*

En bild som visar skärmbild, text, Graf, linje

AI-genererat innehåll kan vara felaktigt.  
*Diagram nr 2: Vanligaste felaktiga prediktionerna i Random Forest.*

En bild som visar skärmbild, text, kvadrat, diagram

AI-genererat innehåll kan vara felaktigt.  
*Bild nr 8: Confusion matrix för Random SVM.*

En bild som visar text, skärmbild, nummer, meny

AI-genererat innehåll kan vara felaktigt.  
*Bild nr 9: Classification Report för Random SVM.*

En bild som visar text, skärmbild, Färggrann

AI-genererat innehåll kan vara felaktigt.  
*Diagram nr 3: Visualisering av klasser i SVM.*

*En bild som visar skärmbild, text, Graf, diagram

AI-genererat innehåll kan vara felaktigt.  
Diagram nr 4: Vanligaste felprediktionerna i SVM.*

# Källförteckning

Statistical Analysis System (2025), Machine learning what it is and why it matters. Hämtad 13 mars 2025 från SAS sida <https://www.sas.com/en_us/insights/analytics/machine-learning.html>

Géron Aurélien, Hands on machine learning with Scikit learn, Keras and Tensorflow, 2:a upplagan, O’Reilly media

IBM, The history of AI, (2024). Hämtad 19 mars 2025 från <https://www.ibm.com/think/topics/history-of-artificial-intelligence>

Kaggle, MNIST in CSV, (2018), hämtad 11 mars 2025, från sidan <https://www.kaggle.com/datasets/oddrationale/mnist-in-csv>

Scikit learn, <https://scikit-learn.org/stable/index.html>

Kaggle, (2018), MNIST Classification using Random Forest, hämtad 16 mars 2025 från <https://www.kaggle.com/code/ashwani07/mnist-classification-using-random-forest>

Medium, (2023), Trying Out Both Random Forest & Decision Tree Classifier On MNIST Using Scikit-Learn, hämtad 16 mars 2025 från   
<https://devkyle4.medium.com/trying-out-both-random-forest-decision-tree-classifier-on-mnist-using-scikit-learn-21c8558c22e6>

Escholarship, (2021), The image classification of MNIST dataset by using machine learning techniques, hämtad 17 mars 2025 från <https://escholarship.org/content/qt0505d532/qt0505d532_noSplash_f17a9f77bcfc62b59306e4196c2cd6dc.pdf?t=r464cp>

Github, (u.å), SVM with MNIST, hämtad 19 mars 2025 från  
<https://dmkothari.github.io/Machine-Learning-Projects/SVM_with_MNIST.html>

Kaggle, (2019), MNIST Digit recognition using SVM, hämtad 19 mars 2025 från  
<https://www.kaggle.com/code/nishan192/mnist-digit-recognition-using-svm>

Pycodemates, (2022), The RBF kernel in SVM: A Complete Guide, hämtad 19 mars 2025 från  
<https://www.pycodemates.com/2022/10/the-rbf-kernel-in-svm-complete-guide.html>

<https://www.geeksforgeeks.org/machine-learning/>

<https://www.datacamp.com/tutorial/svm-classification-scikit-learn-python>

<https://towardsdatascience.com/radial-basis-function-rbf-kernel-the-go-to-kernel-acf0d22c798a/>

<https://www.datacamp.com/tutorial/category/machine-learning>

<https://medium.com/@nik.avgeros/unleashing-the-power-of-ensemble-methods-a-deep-dive-into-classification-using-the-mnist-dataset-1aeefab4c6a6>

<https://dataaspirant.com/k-means-clustering-algorithm/>