Machine learning

Tillämpad MNIST



Daniel Hemgren

EC Utbildning

Machine Learning

20240322

# Abstract

This report describes using the MNIST dataset to train three different models on the MNIST set implemented primarily through the Scikit-learn library. A Streamlit app is developed that can be used for digit prediction. The models are validated to ensure that a good balance between bias and variance is reached. In the next step the models are put into an ensemble voting model to further improve the classification of digits. The major tasks performed in this report are three:

1. An Ensemble model is trained on the MNIST set as described above
2. User written Jpg. digits are transformed to same format as model is trained on (which is array (1,784)
3. A Streamlit App is developed and used to perform predictions on the transformed digits in 2.

Innehållsförteckning

[Abstract 2](#_Toc162032497)

[1 Inledning 4](#_Toc162032498)

[1.1 Syfte 4](#_Toc162032499)

[1.1.1 Frågeställning 1 4](#_Toc162032500)

[1.1.2 Frågeställning 2 4](#_Toc162032501)

[1.1.3 Frågeställning 3 4](#_Toc162032502)

[2 Teori 5](#_Toc162032503)

[2.1 Decision trees 5](#_Toc162032504)

[2.1.1 Random Forest Classifier 5](#_Toc162032505)

[2.1.2 Extra trees classifier 5](#_Toc162032506)

[2.2 Support Vector Machines 6](#_Toc162032507)

[2.2.1 Linjär Support vector klassificering 6](#_Toc162032508)

[3 Metod 7](#_Toc162032509)

[3.1 Data 7](#_Toc162032510)

[3.1.1 T-distributed Stochastic Neighbor Embedding 7](#_Toc162032511)

[3.2 Beskrivning av centrala metodsteg 8](#_Toc162032512)

[4 Resultat och Diskussion 9](#_Toc162032513)

[4.1 Översikt av resultatet och Confusion Matrix 9](#_Toc162032514)

[4.1.1 Resultat Frågeställning 1 10](#_Toc162032515)

[4.1.2 Resultat Frågeställning 2 10](#_Toc162032516)

[4.1.3 Resultat frågeställning 3 11](#_Toc162032517)

[4.2 Diskussion 12](#_Toc162032518)

[5 Slutsatser 13](#_Toc162032519)

[5.1 Frågeställningar 13](#_Toc162032520)

[5.1.1 Svar på Frågeställning 1 13](#_Toc162032521)

[5.1.2 Svar på Frågeställning 2 13](#_Toc162032522)

[5.1.3 Svar på Frågeställning 3 13](#_Toc162032523)

[6 Teoretiska frågor 14](#_Toc162032524)

[7 Självutvärdering 17](#_Toc162032525)

[8 Källförteckning 18](#_Toc162032526)

# Inledning

Computer vision var tillämpningen som fick deep learning att explodera i början av 2010 talet genom tävlingen ImageNet Large Scale Visual Recognition Challenge (ILSVRC) på Stanford. Detta gjorde att äldre tillämpningar som MNIST från 90-talet blev lätt tillgängliga som övningar på nätet genom t.ex Scikit -Learn. Computer vision ger många möjligheter speciellt med ostrukturerade data som nu kan få en struktur via computer vision. (https://image-net.org/challenges/LSVRC/, 2010-2024)

## Syfte

Syftet med denna rapport är att träna modeller på MNIST genomföra träning-validering-test och välja bäst lämpade modell för uppgiften att prediktera vilken handskriven siffra som är på jpg. bilder. Bilden ska kunna laddas upp till den Streamlit app som skapas och som predikterar vilken siffra som är på bilden. Syftet är därmed en tillämpning av computer vision. Bilderna har följande constraints:

* Siffran är skriven med mörk penna på ljus botten
* Bakgrunden har en homogen ljus ton (ej skuggor el. mycket ”Salt&pepper”noise)
* Siffran täcker >50% av bilden i X-led och Y-led
* Siffran är skriven normalt ”rak” ej t.ex kraftigt lutande skrivstil

### Frågeställning 1

Modellen ska uppnå en accuracy > 0,96 med en standardavvikelse över 5 folds av trainsetet på <0,01 dvs en Mratio över 96. Sedan ska också det slutgiltiga Testsetaccuracy i holdout (test) setet ska också vara >0,96.

### Frågeställning 2

Bilder ska transformeras till MNIST standard, dvs (1,784) array att prediktera på

### Frågeställning 3

3 Godkända bilder ska predikteras rätt vid uppladdning till Streamlit appen. minst 2 av 3 bilder ska vara på svåra labels. (Se avsnitt 3.1.1 för definition av svåra labels)

# Teori

Frågeställning1 innebär att mappa bildernas pixlar på tio olika labels och därför är det ett multiklassificeringsproblem. Bland de klassificerare som finns så är decision trees och SupportVectorMachines två olika familjer av klassificerare. (Geron) Inriktningen blir att sätta ihop en ensemble klassificerare där decision trees och SVM kompletterar varandra i modelleringen av datat .

## Decision trees

Decision trees försöker splitta features i homogena delar steg för steg med Gini impurity el entropi som två vanliga mått på denna homogenitet. Denna split pågår recursivt i mindre och mindre sets till alla ”löv” är homogena. (dvs innehåller en label), eller om ett stop krav som tex ”max depth” nås innan. (Geron). En fördel med decision trees är att de är mycket flexibla men det gör också att det är lätt att göra ”overfit” med decision trees och att de sedan inte kan prestera på annan data. Scikit learn använder CART algoritmen som skapar binära träd med två ”grenar” hela tiden. Det finns också andra algoritmer som ID3 men den används inte i denna rapport (Geron, p. 177)

### Random Forest Classifier

Random Forest är en metod som använder sig av randomiserade subsets och skapar en ensemble av decision trees där varje är tränad på ett subset. Subseten väljs med en metod som vanligtvis är ”bagging” och då väljs med återläggning (Geron, p. 193) Metoden tar bort en del av risken som decision trees har i form av overfitting.

”Random forests achieve a reduced variance by combining diverse trees, sometimes at the cost of a slight increase in bias. In practice the variance reduction is often significant hence yielding an overall better model.” (<https://scikit-learn.org/stable/supervised_learning.html>)

Feature importance is easy to measure in random forests. (Geron, p. 198) något som återkommer under avsnitt 3 Metod.

### Extra trees classifier

Extra trees är ”extremely randomized trees” och är ett sätt att ytterligare sänka variance till priset av något högre bias. (Geron, p. 198) Splitting kriteriet väljs inte med ett impurity measure som GINI utan det väljs slumpmässigt. Detta gör att Extra trees går snabbare att träna

” because finding the best possible threshold for each feature at every node is one of the most time-consuming tasks of growing a tree.” (Geron). Extra trees är egentligen en variant av Random Forest som också använder bagging som default för att bygga en skog av decision trees som alla tränas på slumpmässiga delmängder av datat. Resultatet från alla träd kombineras till en förutsägelse. Den slumpmässiga splitten av noder är den centrala unika skillnaden som Extra trees har.

## Support Vector Machines

Support Vector Machines är en kraftfull machine learning modell som skapar ett hyperplan i det flerdimensionella rum som datat kan konceptualiseras som. Hyperplanet kan vara linjärt eller icke-linjärt. Finn det hyperplan som maximerar marginalen är målet för modellen. Att maximera marginalen innebär att hitta det hyperplan som ger det största avståndet mellan supportvektorerna. Supportvektorerna är datapunkterna närmast hyperplanet. Klassificering innebär att titta på vilken sida av hyperplanet som en datapunkt är. SupportVectorMachines har sin största styrka i högdimensionella rum dvs problem med många features.(SVM fungerar även på regression) (Geron, pp. 153-158)

### Linjär Support vector klassificering

LinearSVC är den linjära Supportvectormachine klassificeraren i Scikit learn. Modellen har olika hyperparametrar där C:s invers är proportionell mot nivån av regularisering och är en viktig hyperparameter. (Scikit learn, n.d.). För ändamålet i rapporten så är ett lågt värde på C att föredra då det bör ge bättre generaliserbarhet dvs mindre risk för overfit. Den använder One versus the rest schema för multiklassificering som default och det är lämpligt i detta fall.

# Metod

Miljön som Machine learning projektet görs i är Scikit Learn. Första steget var att hämta MNIST i Scikit learn. Sedan genomfördes för att svara på frågeställning 1:

• Översikts EDA främst TSNE och labelfördelning (utan risk för dataleakage)

• Uppdelning i träning-validering-test set

• Träning av modeller

• Validering av modeller med k fold validation

• Test av modeller i ”verkligheten” (orört holdout/test dataset)

• Exakt kontroll av om målen i frågeställning 1 uppnås

För att svara på frågeställning2:

• Standardisering av mobilbilder till MNIST standard med en serie av transformer.

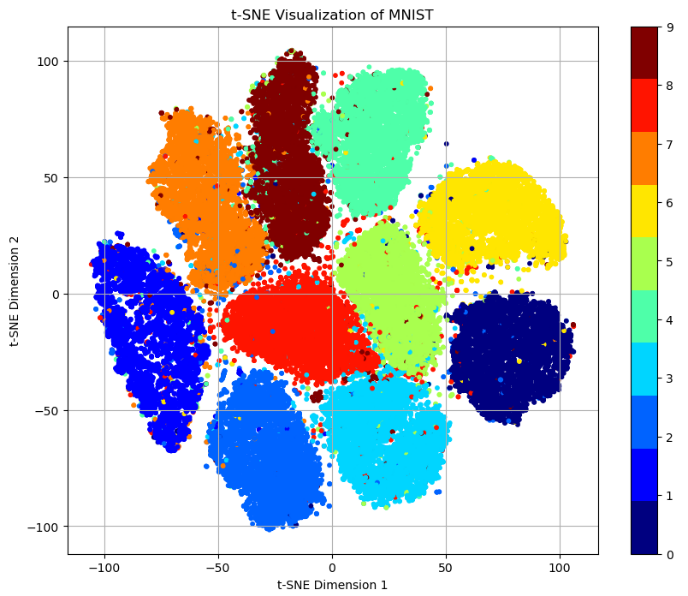
För att svara på frågeställning 3:

• Streamlit applikation skapas och prediktion sker av 3 godkända bilder.

## Data

MNIST är ett standardiserat dataset som består av 70000 handskrivna siffror där alla har labels 0-9. (http://yann.lecun.org, n.d.) Datasetet har en nästan jämn fördelning av labels (det är viktigt gällande t.ex class-weight hyperparametern i LinearSVC där default är just jämn fördelning). Datasetet har väldigt lågt random noise då det är standardiserat. (http://yann.lecun.org, n.d.)

### T-distributed Stochastic Neighbor Embedding

Nedan är sklearn.manifold.TSNE funktionen använd på MNIST datat och labels blir visualiserade i relation till varandra. Här kan vi se att t.ex 8 och 5 ligger i mitten och är rimligtvis relativt de andra svåra labels att prediktera oavsett modell. Dvs 1 är troligtvis en ”enkel” label att modellera, då den ligger isolerad till vänster med nästan ingen overlap till andra labels i y-led på denna TSNE , ett litet overlap till 7 finns men på långt avstånd.

Figur 1: TSNE applicerad på MNIST

## Beskrivning av centrala metodsteg

Testdelen av datat skiljs av direkt i en 60000-10000 split. Sedan kan de 60000 användas för träning och validering utan risk för data leakage.

En hard voting modell skapas som använder hard voting med de tre klassificerarna RandomForest, ExtraTrees och LinearSVC som input till votingmodellen. Voting är för att minimera risken för overfitting då jag hellre tränar fram en lite mer biased modell som är konsekvent även utanför MNIST setet.Om datat har mycket noise så kan Voting ibland inte minmera overfit (http://yann.lecun.org, n.d.) MNIST setet har lågt noise så därför är det ett lämpligt val.

Att t.ex göra Gridsearch på Hyperparametrarna i SVC hade kunnat ge något lägre bias men höjer risken för overfit. Jag väljer att prioritera låg variance och tar då hellre något högre bias , ”all models are wrong but some are useful” (Box).

De tre modellerna tränas och sedan genomförs k fold validation med 5 stratified folds i train setet. Slutligen görs kontroll av accuracy i Valideringssetet och sedan görs slutgiltig kontroll i testsetet. De modeller som godkänns får ingå i ensemble voting classifiern. Voting classifiern genomgår sedan ännu en gång samma steg som varje individuell modell. Kontroll av om resultatkraven i frågeställning1 uppnås görs. Resultatet ses i kapitel 4.

Frågeställning2 inleds med att först samla in några transformer t.ex transform.resize , color.rgb2gray, util.img\_as\_ubyte, StandardScaler, .flatten, (Scikit learn, n.d.) invertering . Dessa testas med ibland inget bra resultat och ibland en förbättring till slut hittas en ordning som fungerar.

Frågeställning3 inleddes direkt och ett embryo till Streamlit app sattes upp i ett tidigt skede för att se vidden av hela uppgiften ”från ax till limpa”. Streamlit appen har sedan använts för att prediktera olika handskrivna siffror. Främst siffror som är inom de constraints som sattes i syftesdelen.

# Resultat och Diskussion

## Översikt av resultatet och Confusion Matrix

Extra trees är den bästa estimatorn och slår faktiskt voting klassificeraren i accuracy. (den blir sannolikt bortröstad av de två lite svagare estimatorerna ibland och därmed kan voting estimatorn bli något svagare). I detta läge valde jag ändå att använda voting estimatorn då den bör uppvisa lägre varians på andra typer av siffror.

Confusion Matrix är balanserad och ser ut enligt nedan för Votingclassifiern på testsetet:

Det som överraskar mest är att sanna 4:or predikteras som 9:or vid 22 tillfällen. (Av ca 900 sanna 4:or är det en acceptabel avvikelse. Det är också förståeligt vid en titt på TSNE)

A chart with numbers and labels

Description automatically generated

Figur 2: Confusion Matrix på testsetet (10000 av totala 70000 i MNIST)

### Resultat Frågeställning 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Kfold cross validation Mean accuracy stdev Mratio** | | | |
| RandomForestClassifier | 0,9661 | 0,0018 | 536,7 |
| ExtraTreesClassifier | 0,9695 | 0,0007 | 1385,0 |
| LinearSVC | 0,8537 | 0,0065 | 131,3 |
|  |  | | |
| VotingClassifier (hard voting used) | 0,9659 | 0,0012 | 804,9 |

Tabell 1: Kfold cross validation Mean accuracy score och accuracy stdev och Mratio. Min egen ModelstabilityRatio (Mratio) är en analogi till Sharpe ratio inom kapitalförvaltning där målet är högt medelvärde med låg varians.

|  |  |
| --- | --- |
| **Accuracy score i validation set för olika modeller** | |
| RandomForestClassifier | 0,9692 |
| ExtraTreesClassifier | 0,9715 |
| LinearSVC | 0,8590 |
|  |  |
| VotingClassifier (hard voting used) | 0,9693 |

Tabell 2: Accuracy i validation set

|  |  |
| --- | --- |
| **Accuracy score i testsetet för olika modeller** | |
| RandomForestClassifier | 0,9645 |
| ExtraTreesClassifier | 0,9691 |
| LinearSVC | 0,8566 |
|  |  |
| VotingClassifier (hard voting used) | 0,9650 |

Tabell 3: Accuracy i testset

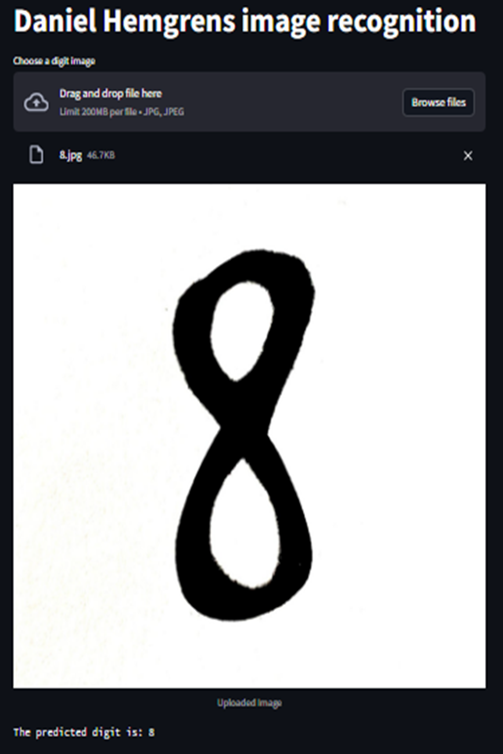
### Resultat Frågeställning 2

Transformer till rätt storlek , gråskala, whitening, normaliserng, flattening och invertering gjorde att bilden fick rätt format att prediktera på. Bilden kan därmed användas av votingmodellen.

### Resultat frågeställning 3

Här ses resultat för att svara på frågeställning3: 5 och 8 ovan är svåra labels och rätt prediktion ges av Streamlit appen som använder Voting classifiern. Även prediktionen av 2:an är korrekt.

A number on a piece of paper

Description automatically generated

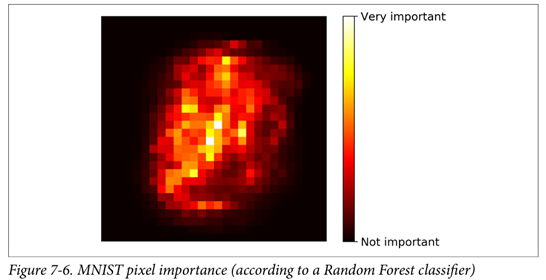
A black line on a white paper

Description automatically generated

## Diskussion

Resultaten har överlag varit bra och det är förmodligen delvis för att MNIST är ett standardiserat set. För att skapa en mycket robust modell kan man tänka sig att föra in ”noise” i MNIST setet innan träning att ”vrida och vända förstora och förminska” redan innan träning. Det är en metod som används inom Convolutional Neural Networks om jag förstått det hela rätt. Det kan bli intressant till Deep learning kursen.

Bilden nedan (Geron, p. 199) fångade min uppmärksamhet tidigt och den sätter igång tankarna för hur man kan tänka sig att en kraftig ”feature reduction” (där endast gula och ljusa features nedan viktas upp) hade kunnat skapa en ”genväg genom MNIST”. Efter att sökt information om problemet så valde jag ändå att gå vidare med modellering av alla 784 features. Detta för att MNIST är inget jätteset och det jag kunnat vinna i feature reduction hade jag förlorat i information loss. Hela denna aspekt är mer relevant när det gäller vädigt stora set.



# Slutsatser

Den generella slutsatsen är att det har varit ett intressant och roligt arbete där modeller tränats på MNIST och handskrivna bilder transformerats för att slutligen bli input till prediktioner i den utvecklade Streamlitappen.

## Frågeställningar

De tre frågeställningarna i Syftesdelen kommer här gås igenom för att se om frågeställningarna kunde besvaras.

### Svar på Frågeställning 1

”Modellen ska uppnå en accuracy > 0,96 med en standardavvikelse över 5 folds av trainsetet på <0,01 dvs en Mratio över 96. Sedan ska också det slutgiltiga Testsetaccuracy i holdout (test) setet ska också vara >0,96.”

* Votingmodellen uppnår en Mratio på 804,9 vilket är över målet 96.
* Votingmodellen uppnår en accuracy i testsetet på 0,965 vilket är över målet 0,96

Frågeställning 1 är besvarad.

### Svar på Frågeställning 2

”Bilder ska transformeras till MNIST standard, dvs (1,784) array att prediktera på”

* Bilder transformerades i flera steg till formatet (1,784) som fungerade för prediktion. Frågeställning 2 är besvarad.

### Svar på Frågeställning 3

”3 Godkända bilder ska predikteras rätt vid uppladdning till Streamlit appen. minst 2 av 3 bilder ska vara på svåra labels. (Se avsnitt 3.1.1 för definition av svåra labels)”

* 3 Godkända bilder varav två svåra(5 och 8) predikterades korrekt av den utvecklade Streamlit appen. Frågeställning 3 är besvarad.

# Teoretiska frågor

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

Träning är det största datasetet och är det första Kalle bör träna sina modeller på. Med Valideringsdatan kan Kalle se om modellen fungerar även utanför träningssetet och om inte så är förmodligen modellen ”overfitted” på träningsdatat då bör hyperparametrar korrigeras. Testdatat ska vara orört och fungera som ”verkligheten” här testas om modellen verkligen fungerar på riktigt som sista steg.

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 använda k-fold cross validation och träna på subsets av träningsdatan. Den modell som uppvisar bäst resultat med k-fold cross validation bör hon fortsätta med. Dvs modellen med hög accuracy som har låg varians över olika folds.

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

potentiella tillämpningsområden?

Ett regressionsproblem handlar om att vår frågeställning är kontinuerlig. Det kan vara t.ex Huspriser som predikteras med features såsom Inkomst, GPS position, antal rum etc. Linjär Regression är en vanlig modell. SVM och decision trees kan också användas om vår frågeställning är kontinuerlig.

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

RMSE är roten ur MSE och kan tolkas i relation till vår frågeställning (y).

Om y är t.ex huspriser på USD 300000 så kan en RMSE på 70000 ställas i relation till 300000 som ett mått på storleken av modellens normala fel.

(RMSE kan jämföras med Standardavvikelse i benämningen att båda är jämförbara med våra målvärden. (MSE och Varians är inte lika lätt jämförbara med våra målvärden))

5. Vad är ”klassificieringsproblem? Kan du ge några exempel på modeller som används och

potentiella tillämpningsområden? Vad är en ”Confusion Matrix”?

Logistic regression är ett exempel på en linjär klassificerare. Ett exempel är en kund som har vissa features som vi kan modellera till att kategorisera kunden som en ”churn” eller ”inte churn”. ”Confusion Matrix” delar upp felen i precision och recall. Där det alltid blir en avvägning. Exempel är rättegång där samhället prioriterar precision och det innebär att de dömda är skyldiga väldigt ofta. Baksidan är en något sämre recall där skyldiga kan gå fria.

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

Det är en unsupervised clustering algoritm. Modellen kräver inga labels och kan användas t.ex på kunder som rör sig i ett varuhus. De som är närmast varandra i hur de rör sig i varuhuset är med K-means metoden i samma ”kundgrupp” t.ex. (Modellen är unsupervised för att inga uppdelningar av kunder görs innan t.ex varken ålder/kön/annat. Datat saknar ”labels” att träna mot)

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 är ofta med integers där vi vill använda en ordinalitet på datan. One-hot encoding är med columner och 1 för värdet och 0 annars. Dummy variable encoding är som one-hot encoding men utan redundans, den är också ett krav för modeller som använder linjär regression( och andra regressions modeller som har en bias term).

* Väder Ordinalt: Molnigt-halvklart-sol (1-2-3 som koder blir ordinal)
* Väder One-hot: tre kolumner(Molnigt-halvklart-sol) 1 el 0 blir one-hot
* Väder Dummy Variable: Två kolumner (Halvklart-Sol) som ovan och 0-0 är Molnigt.

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?

Om kategorier är ordinala eller inte kan väljas av den som kodar dem. Vill vi t.ex räkna median så behöver vi en ordinalitet. Julia har rätt, så länge kategorier finns så kan ordningen tänkas betyda något. (I fallet färger har de t.ex också olika frekvens i elektromagnetisk mening. )

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 en open-source miljö för app utveckling inom data science. Det är främst en hjälp till front end utvecklingen av appen. Det kan användas för att skapa t.ex interaktivitet där användare kan kategorisera t.ex bilder som katt/inte katt.

# Självutvärdering

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

Jag var medveten om att transformeringen av mobilbilderna är en utmaning och därför låg mitt fokus här . Efter en del experimenterande med ”greyscale”, pixelstandarder, flattening,reshaping funktioner och annat så fungerade det inte ändå. Jag tog och tittade på mina 28\*28 matriser ordentligt och kom fram till att en invertering kan göra att modellen hittar rätt. Mycket riktigt så var invertering av datan det som gjorde att modellen hittade rätt siffra i mönstret av de 784 talen. Det var riktigt roligt när jag insåg varför! (Därför har min sista konvertering av arrayen variabelnamnet ”inv” så jag kommer ihåg att det fungerar som sista steg åtminstone i detta exempel.)

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

Jag tycker att jag förstår området och har löst de utmaningar jag stött på och är värd VG. Att få det hela att lira i Streamlit var en stor utmaning då det är nytt hur ”appdevelopment” går till. Streamlit är ett mycket bra verktyg.

1. Något du vill lyfta fram till Antonio?

Jag har väntat på när vi ska använda ”riktig data” och detta var verkligen riktig data och mycket intressant. För egen del så är jag van med data från finansmarknaden(din master om diversifiering var mycket intressant) och hur den kan se ut men andra typer (som t.ex pixelbilder) är mycket intressanta och jag vill se mer av olika typer av data. Jag har en känsla av att om man abstraherar olika datatyper så borde de vara ”samma sak” en bit under ytan. Det är mitt mål på några års sikt: att lära mig data transformations så att vad som helst blir ”apples to apples”.

# Källförteckning

Box, G. (u.d.).

Geron, A. (u.d.). *(Hands-On\_Machine\_Learning\_with\_Scikit-Learn-Keras-and-TensorFlow-2nd-Edition-Aurelien-Geron.pdf).*

*http://yann.lecun.org*. (u.d.).

*https://image-net.org/challenges/LSVRC/*. (2010-2024).

*https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#sklearn.ensemble.RandomForestClassifier*. (u.d.).

*https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC*. (u.d.).

*https://scikit-learn.org/stable/supervised\_learning.html.* (u.d.).

*Scikit learn*. (u.d.). Hämtat från https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC.