Ensemble Learning MNIST

Ensemble (4x) *vs.* Single Modell (1x)



Siarhei Thor

EC Utbildning

Machine Learning 01

202402

# Abstract

This report showcases a comprehensive Machine Learning project flow, centered around the MNIST dataset, utilizing the scikit-learn library in Python to develop a digit recognizer Streamlit app. The study emphasizes the effectiveness of ensemble techniques like Voting Classifier and Stacking Classifier alongside hyperparameter tuning through grid search to enhance different models performance. However, it was discovered that while ensemble methods significantly increased model size, a single SVC model, when combined with PCA for dimensionality reduction, not only reduced the model size to a manageable 15MB but also slightly outperformed the more complex ensemble approaches. The report concludes that simpler models can be more efficient and cost-effective, especially in cloud deployments, highlighting the critical balance between model complexity and operational practicality.

Repo: [https://github.com/SiarheiThor/ensemble-learning-sklearn](https://github.com/SiarheiThor/ensemble-learning-sklearn/tree/main)

App: <https://ensemble-learning-sklearn-fhregmhvioxh7lmvzuwcm6.streamlit.app/>



Innehållsförteckning

[Abstract 2](#_gjdgxs)

[1 Inledning 1](#_3znysh7)

[2 Teori 2](#_tyjcwt)

[3 Metod 4](#_3rdcrjn)

[Process Flödet 4](#_1hk90t1z7mq5)

[Data 4](#_wunoqmwe9f83)

[Modeller 4](#_w6c5wvsanmml)

[Hyperparameter Tuning med Grid Search. 4](#_9u0dli82bnet)

[Ensemble-metoder 5](#_d875y3tpzc6v)

[Dimensionalitetsreduktion med PCA 5](#_zhe0lu5mtrxo)

[Data Augmentering 5](#_ozrro0ecn64i)

[4 Resultat och Diskussion 6](#_26in1rg)

[5 Slutsatser 8](#_lnxbz9)

[6 Teoretiska frågor 9](#_35nkun2)

[7 Självutvärdering 11](#_1ksv4uv)

[Källförteckning 12](#_2jxsxqh)

# Inledning

Denna rapport är en slutleverans inom kunskapskontroll 2 för Machine Learning. Syftet är att vissa kompetens inom det typiska ML projekt flödet, med ett pilotprojekt Digit Recognizer, ett streamlit app som predikterar siffra på uppladdat bild, med hjälp av SKlearn biliotek för python. Rapporten kommer även innehålla den teoretiska delen . För att uppfylla syftet så kommer följande frågeställning(ar) att besvaras:

1. Teoretiska frågor kopplade till ML kurs.
2. Teoribeskrivning för projektet.
3. Metodbeskrivning
4. Resultat & Diskussion
5. Slutsatser

## 

# Teori

Teoridelen beskriver alla koncept som har använts i projektet.

RandomForestClassifier

RandomForestClassifier är en ensemble-lärandemetod som använder flera beslutsträd för att utföra klassificering. För att skapa mångfald och minska risken för överanpassning, använder den tekniker som bootstrap sampling (att skapa olika träningssubset med ersättning) och feature randomness (vid varje split, väljer en slumpmässig delmängd av features). Det slutliga beslutet görs genom majoritetsröstning från alla träd i skogen, vilket ofta leder till hög prestanda och robusthet mot överanpassning.

ExtraTreesClassifier

ExtraTreesClassifier (Extremely Randomized Trees) är lik Random Forest men med ännu mer slumpmässighet i konstruktionen av träden. Här görs delningen helt på slumpmässiga värden för varje feature snarare än att söka efter den bästa möjliga spliten. Detta kan leda till en snabbare träningsprocess och i vissa fall bättre modellgeneralisering, eftersom det minskar modellens varians ännu mer.

SVC (Support Vector Classifier)

SVC (Support Vector Classifier) är en del av support vector machines (SVM) och används för klassificering. Den fungerar genom att hitta den hyperplan som bäst separerar de olika klasserna i datan. SVC kan använda olika kernel-funktioner för att hantera icke-linjär separerbarhet genom att projicera data till högre dimensioner där de kan separeras linjärt.

SKlearn Pipeline (StandardScaler och PCA)

En Pipeline i scikit-learn är en kedja av transformerande steg med en avslutande estimator. I kontexten av en StandardScaler och PCA (Principal Component Analysis) pipeline, först standardiserar StandardScaler datan (ger den medelvärde 0 och varians 1) för att undvika bias av features med olika storleksordningar. Därefter reducerar PCA datans dimensioner genom att projicera den till en mindre dimensionell rymd samtidigt som så mycket information som möjligt bevaras, vilket ofta hjälper till att förbättra modellens effektivitet och minska överanpassning.

KNeighborsClassifier

KNeighborsClassifier är en typ av lata inlärningsalgoritm (lazy learning algorithm) och tillhör kategorin övervakad maskininlärning. Den klassificerar datapunkter baserat på de 'k' närmaste grannarnas klasser. Algoritmen kräver ingen träning för datamodellering men behöver hela datasetet för att göra förutsägelser, vilket gör den resurskrävande vid stora datamängder.

Hyperparameter

Hyperparametrar är de parametrar som inte lärs av modellen under träningsprocessen. De måste ställas in innan träningen startar. Exempel på hyperparametrar inkluderar antalet träd i en RandomForestClassifier, C och gamma i en SVC, och antalet grannar k i en KNeighborsClassifier. Att välja rätt hyperparametrar är avgörande för modellens prestanda.

Grid Search

Grid Search är en metod för hyperparameteroptimering som systematiskt itererar igenom en fördefinierad lista av hyperparametrar, tränar modellen med varje kombination och utvärderar dess prestanda. Det hjälper till att hitta den bästa uppsättningen hyperparametrar för en given modell.

Voting Classifier

Voting Classifier är en ensemble-metod i scikit-learn som kombinerar olika modeller och använder majoritetsröstning (för hård röstning) eller viktad medelvärde (för mjuk röstning) för att göra slutgiltiga förutsägelser. Detta kan ofta leda till bättre prestanda jämfört med enskilda modeller.

Stacking Classifier

Stacking Classifier är en annan ensemble-metod som staplar olika modeller. Istället för att rösta direkt, tränas en meta-modell på outputen från de basmodeller som ingår. Detta innebär att de ursprungliga modellernas förutsägelser används som input för att träna en ytterligare modell (meta-learner) som sedan gör den slutgiltiga prediktioner. Detta teknik kan leda till mer avancerade sammanvägningar av information från de olika modellerna och potentiellt bättre prestanda.

Data Augmentation

Data augmentation är en teknik för att öka mångfalden i ett dataset utan att faktiskt samla in fler data. Genom att tillämpa en rad slumpmässiga transformationer på befintliga bilder (som rotation, skalning, skärning, eller tillägg av brus) kan man skapa nya träningsdata. Denna metod är särskilt populär inom bildigenkänning och hjälper till att förbättra modellernas generaliseringsförmåga och robusthet genom att simulera olika perspektiv och villkor som modellen kan stöta på i verkliga scenarier.

## 

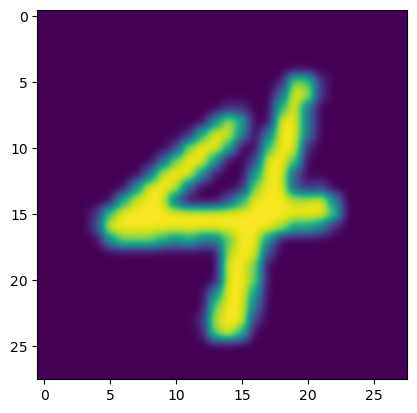
# Metod

## Process Flödet

* Samla Data (MNIST är inte svårt att hämta in )
* Dela datan i train, val, test
* Analysera train data
* Välja modellen baserad på analys av train data
* Grid search för hyper parameter tuning med train + test och korsvalidering
* Utvärdering av alla modeler, med val och test data
* Slutlig träning av den best modell med all data
* Kod refactoring för distribuering
* Lokala tester av app
* Förbättringar (I min fal - data augmentation)
* Distribuering med Streamlit

## Data

MNIST databasen, som innehåller handskrivna siffror, är en populär resurs för att träna och testa bild klassificeringsmodeller.



## Modeller

Fyra olika klassificeringsmodeller användes: RandomForestClassifier, ExtraTreesClassifier, SVC (Support Vector Classifier) inom en Pipeline för att standardisera data, och KNeighborsClassifier. Dessa modeller valdes baserat på deras förmåga att hantera högdimensionella data och deras effektivitet i klassificeringsuppgifter.

## Hyperparameter Tuning med Grid Search.

För att hitta de bästa hyperparametrarna för varje modell genomfördes en grid search över ett definierat parameterutrymme. Denna process optimerar modellens prestanda genom att iterativt testa olika kombinationer av parametrar.

Best parameters for RandomForestClassifier: {'max\_depth': None, 'min\_samples\_split': 2, 'n\_estimators': 200}

Best score for RandomForestClassifier: 0.9672833333333334

Fitting 3 folds for each of 12 candidates, totalling 36 fits

Best parameters for ExtraTreesClassifier: {'max\_depth': 50, 'min\_samples\_split': 2, 'n\_estimators': 200}

Best score for ExtraTreesClassifier: 0.97125

Fitting 3 folds for each of 3 candidates, totalling 9 fits

Best parameters for Pipeline: {'svc\_\_C': 10, 'svc\_\_kernel': 'rbf'}

Best score for Pipeline: 0.9682166666666667

Best parameters for KNeighborsClassifier: {'algorithm': 'ball\_tree', 'n\_neighbors': 3, 'weights': 'distance'}

Best score for KNeighborsClassifier: 0.9702166666666666

## Ensemble-metoder

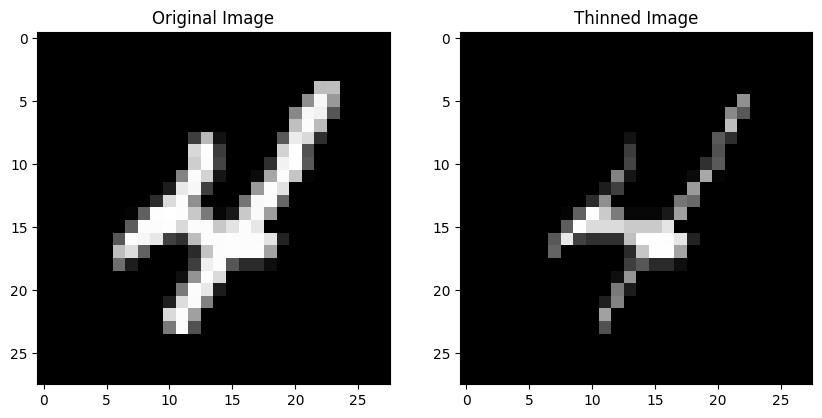
Två ensemble-tekniker användes: hård röstning (VotingClassifier) och stacking (StackingClassifier). Ensemble-metoder kombinerar flera modeller för att förbättra den totala klassificeringsprestandan genom att dra nytta av styrkorna hos varje individuell modell.

## Dimensionalitetsreduktion med PCA

PCA användes för att minska dimensionerna av data innan modellträningen. Detta hjälper till att minska överanpassning och beräkningstid, samt möjliggör användning av mindre och mer hanterliga modellstorlekar.

## Data Augmentering

För att förbättra klassificeringsprestandan för bilder med tunnare skrivna siffror implementerades data augmentering genom att justera tjockleken på siffrorna. Detta hjälper modellen att bli mer robust mot variationer i datan.



# Resultat och Diskussion

För att distribuera appen bör modellfilen vara så liten som möjligt utan att förlora kvalitet. När ensembletekniker testades blev den totala storleken på modellen 2,1 GB, vilket är alltför stort för denna enkla applikation.

**Accuracy för modeller**

|  | Train/Val | Test |
| --- | --- | --- |
| Hard Voting with PCA on all 4 models  (Random Forest, Extra Trees, SVM, KNN, Hard Voting) | 0.8101, 0.9728, 0.9707, 0.9714, 0.9753 | 0.9727 |
| Modell Stacking | - | 0.9775 |

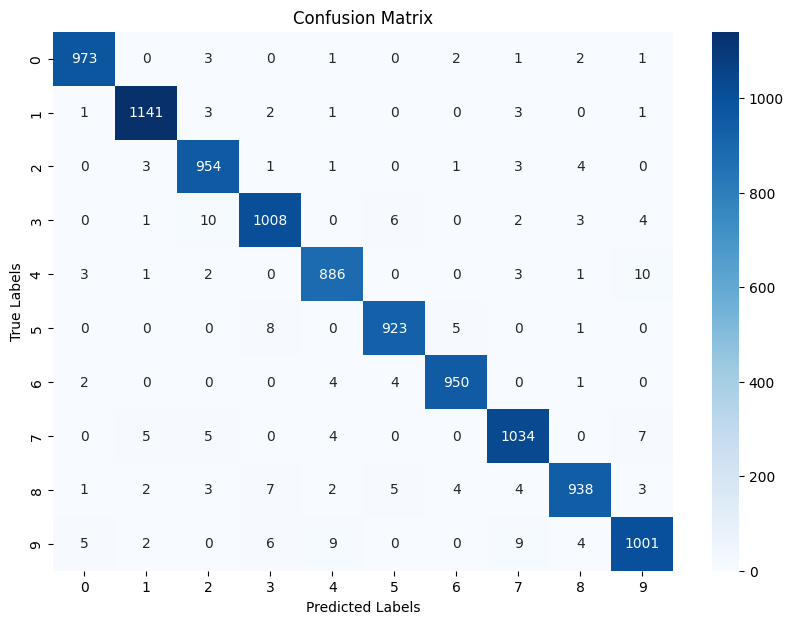
Därför har jag testat alternativ, och PCA visade sig vara ett bra knep för att minska storleken till endast 15 MB för den enskilda SVC-modellen. Förvånansvärt nog presterade den till och med lite bättre än båda modellensemblerna.

**Accuracy för modeller med PCA (95%)**

|  | Train/Val | Test |
| --- | --- | --- |
| Hard Voting with PCA on all 4 models  (Random Forest, Extra Trees, SVM, KNN, Hard Voting) | 0.9497, 0.9524, 0.9871, 0.9726, 0.9695 | 0.9666 |
| Modell Stacking | - | 0.9817 |
| Single SVC | 0.982 | 0.982 |

**Less is more!**

Single SVM with PCA Confusion matrix:



*Figur 1: Confusion matrix för Single SVC model.*

# Slutsatser

Ensemble tekniker är ett väldigt kraftfull verktyg, dock ska det användes sparsamt där det är verkligen behövs. Efter PCA dimensions reducering ett enkel SVC har visat sig att vara lika högpresterande som ett tyng Stacking Classifier, sparade modelsorlekar: 15 mb vs 1 gb. Mindre modeller är lättare att hantera i cloud, snabbare att träna och darmed kostnadseffektiva. MNIST är väldigt ren och enkel dataset där enkel modell presterar bättre än ensembler.

Några övriga punkter:

**Ensemble-metoder**: Användningen av VotingClassifier och StackingClassifier framhäver hur kombinationen av modeller kan förbättra noggrannheten.

**PCA Effektivitet**: PCA minskar modellstorleken effektivt utan att förlora prestanda, vilket är kritiskt för distribution.

**Data Augmenterings:** Projektet visar hur data augmentering ökar modellernas generaliseringsförmåga, speciellt för specifika utmaningar som tunnare siffror.

# Teoretiska frågor

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

*Train - träna modellen.*

Validation - kontrollera modellens prestanda, utvärdera hyper parametrar.

Test - används för att granska resultatet av slutmodel, så nära äkta scenariot som möjligt. används ofta med att granska prestanda av final estimator i kontext av ensemble learning.

1. *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”?*

Använd korsvalidering.

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

Regressionproblem är att prediktera kontinuerlig variabel baserad på ett antal oberoende variabler. Exempel: predektera huspriser baserad på läge, storlek, antal rum, mm.

1. *Hur kan du tolka RMSE och vad används det til?*

RMSE är ett mått på hur bra modeller kan prediktera. Större avvikelser mot äkta värden påverkar måttet mer än till exempel MAE. Används framförallt för modellprestanda validering. Rekomenderas dock att använda fler mått i kombination med RMSE som MAE och r2.

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

klassificieringsproblem är att predektera kategoriska utfall baserad på oberoende variabler. Spamdetektering är ett av mest kända exempel på klassificieringsproblem. Confusion matrix är ett sätt att sammanställa antal True Positives, False Positives, True Negatives, False Negatives. Med dessa värden kan vi ta fram prestandamått som t.ex. precision, recall, F1-score, mm.

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

Är ett klustering modell som används oftast för ‘unsupervised learning’ (där vi inte har kända utfäll). Används inom t.ex marknadssegmentering för att skapa klusters eller grupper för till exempel kund beteende för att optimera marketingkampanjer.

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

Nämnda tekniker används för att kunna jobba med kategoriska variabler med algoritmer som inte kan hantera kategoriska variabler.

1. *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. Om data är ordinal eller nominal beror på kontexten av problemställning. I vårt exempel färgerna allmänt är nominal, men i kontext av skjorta kan vissa uppleva som vackrare än andra och därmed har en ranking eller ordning.

1. *Vad är Streamlit för något och vad kan det användas till?*

Streamlit är ett open source bibliotek för Python som används för att skapa webbapplikationer för maskininlärning och dataanalys med minimal kodning.

# Självutvärdering

1. Utmaningar du haft under arbetet samt hur du hanterat dem.   
   Skillnaden mellan tränings dataset och real data är för stor för att kunna skapa en bra app, augmentering har förbättrad prestanda något men modellen har fortfarande svårt med tun skrift och över stora siffror.
2. Vilket betyg du anser att du skall ha och varför.   
   VG alla kriterier uppfyllda.
3. Något du vill lyfta fram till Antonio?

En diskussion kring när kombinerade modeller kan signifikant höga prestanda jämfört med en bra enkel modell.

# 

# Källförteckning

Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (2nd ed.). O'Reilly Media.