

Handskriven Sifferigenkänning med Maskininläring



Priyadarsini Panda

EC Utbildning

Examensarbete

2025-03

Abstract

This Project aims to develop and evaluate a machine learning model for digit recognition using the MNIST data set. The study focuses on applying machine learning techniques to preprocess and analyze hand written digit data for accurate classification. The methodology emphasizes achieving high performance by addressing challenges such as hand written variation and image noise. The model is designed and aiming to achive at least 95% accuracy, showcasing its potential for real world application. This research highlights uses of digit recognition in banking, vehicle licence plate identification, digitization of hand filled forms.

Keywords: Digit Recognition, Image Processing, MNIST Dataset, Hyperparameter Tuning, Model Training, Model Evaluation, Performance Metrics, Confusion Matrix, Joblib(model persistence).

Acknowledgement

I would like to express my deepest gratitude to my family and friends for their unwavering support, encouragement , and belief in me.

I am also immensely thankful to my Mentors Linus and Antonio for their invaluable guidance and expertise , which have been a constant source of inspiration and learning. Their insightful teaching have shaped my understanding and enriched my growth.

Lastly, a heartfelt thanks to my classmates for their kindness and generosity.

Contents

Abstract	2
1 Inledning.....	5
1.1 Syfte	5
1.2 Frågor	5
Outline Of the text.....	6
2 Teori 2.1 Dataöversikt	7
2.2.1 Logistic Regression	7
2.2.2 Random Forest	8
2.2.3 Support Vector Machine	8
2.3 Model Evaluation Framework.....	9
2.3.1 Confusion Matrix	9
2.3.2 HyperParameter	10
3. Metod	11
3.1.1 Feature Scaling	11
3.1.2 Dimensionality Reduction.....	11
3.1.3 Model Training.....	11
3.1.4 Model Evaluation.....	11
4. Resultat och Performance	12
4.1 Support Vector Machine (SVM).....	12
4.2 Logistic Regression	12
4.3 Random Forest	12
5. Slutsats	13
Part 2. Teoretiska frågor.....	14
2 Självtvärdering.....	18
Appendix A	19
Källförteckning.....	20

1 Inledning

Handskrivna siffror har länge varit ett grundläggande problem inom området datorseende och maskininlärning. Möjligheten att känna igen handskrivna siffror är inte bara kritisk utan fungerar också som ett benchmarkproblem för att testa och jämföra olika maskininlärningsalgoritmer. MNIST-datauppsättningen, bestående av 28x28 gråskalebild av handskriven siffra. Den är utformad för att hjälpa forskare att utveckla och testa maskininlärningsalgoritmer inom mönsterigenkänning och maskininlärning. Datauppsättningen hittar applikationer inom banksektorn, posttjänster och dokument hantering. Genom att utforska och jämföra tillvägagångssätt som Logistic Regression, Random Forest och Support Vector Machines. Denna rapport syftar till att identifiera den mest effektiva metoden för igenkänning av handskrivna siffror.

1.1 Syfte

Syftet med denna rapport är att undersöka och utforska effektiviteten av maskininlärningstekniker för igenkänning av handskrivna siffror och utvärdera deras prestanda. Genom att jämföra modellens prestanda strävar vi inte bara efter att förstå deras styrkor och begränsningar i denna uppgift utan bidrar också till gränsförståelsen av bildklassificeringstekniker inom maskininlärning.

1.2 Frågor

För att uppfylla detta syfte kommer följande frågor att besvaras:

- Vilka förbearbetningssteg och feature extraktions metoder förbättrar prestandan för sifferigenkänning?
- Vilken maskininlärningsmodell (logistisk regression, Random Forest eller SVM) ger mer än 90% accuracy?

Outline Of the text

Denna rapport börjar med en introduktion av handskriven sifferigenkänning och dess betydelse. Den beskriver sedan den teoretiska bakgrunden till Logistic regression, Random Forest och SVM som är relevanta för att förstå de tekniker som används. Därefter beskriver den förbehandlingsstegen, experimenten och metoderna som tillämpas i denna studie. Därefter presenteras och analyseras resultaten utifrån modellernas prestanda. Slutligen avslutas rapporten med en sammanfattning av resultat och insikter från studien.

2 Teori

2.1 Dataöversikt

MNIST dataset består av gråskalebilder av handskrivna siffror som sträcker sig från 0 till 9. Varje bild representeras som rutnät på 28x28 pixlar. Datasetet innehåller 60,000 bilder som träningsset och 10,000 bilder som testset. Siffrorna gör datasetet väl lämpat för utvärdering och benchmarking av maskininlärningsmodeller.

2.2 Models Use

Här har vi att göra med klassificeringsproblemet. I detta forskningsarbete kommer vi att använda logistisk regression, Random Forest och Support Vector Machines (SVM) som är de vanligaste klassificerarna inom maskininläring. Detta avsnitt syftar till att ge en allmän översikt av teori bakom olika modeller.

2.2.1 Logistic Regression

Logistisk regression är en binär klassificerare som uppskattar sannolikheten för att en händelse inträffar med hjälp av logistisk funktion.

$$f(x) = \frac{1}{1+e^{-x}}$$

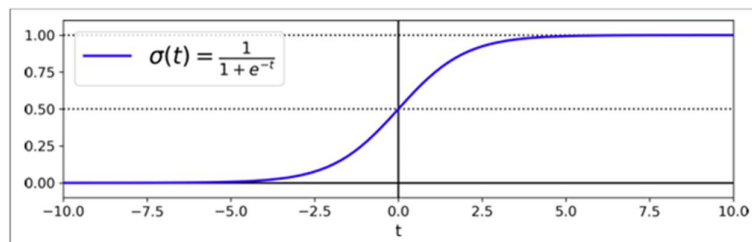


Figure 1. Logistic function

(Image taken from (p.143) in the second edition of Hands on machine Learning with Scikit-Learn Keras & Tensorflow, Aurélien Géron.)

Funktionen ger en S-formad kurva. Den här kurvans svaga lutning möjliggör mjuk övergång, vilket gör att den hanterar klassificeringsproblem.

2.2.2 Random Forest

Random Forest är en maskininlärningsalgoritm som används för klassificerings och regressionsuppgifter. Det hjälper till att prediktera resultat genom att kombinera resultaten från flera beslutsträd. Random Forest ger mycket exakta förutsägelser även med stora datamängder.

Den kan hantera saknade data bra utan att kompromissa med accuracy. Det kräver ingen normalisering eller standardisering på datasetet och när vi kombinerar flera beslutsträd minskar risken för överanpassning av modellen.

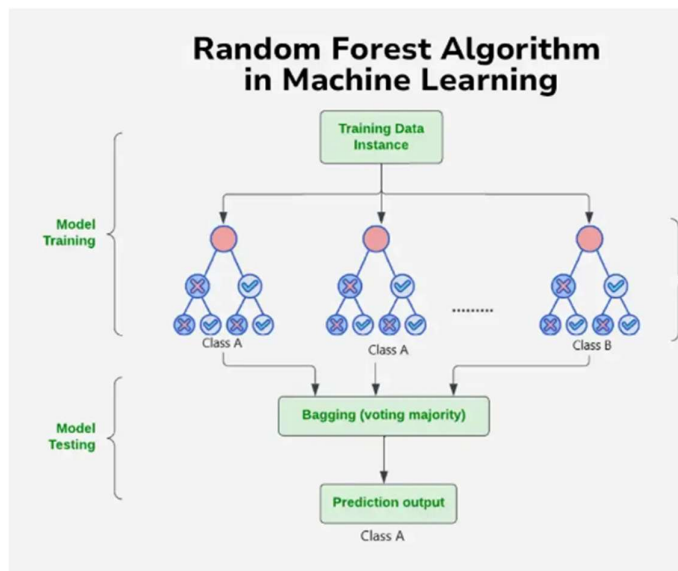


Figure 2. Random Forest
(Image taken from greeksforgeeks.org)

2.2.3 Support Vector Machine

Support Vector Machines (SVM) är en kraftfull maskininlärningsalgoritm som används för klassificerings- och regressionsproblem. SVM passar bra för små eller medelstora datamängder. Det fungerar genom att hitta det optimala hyperplanet som bäst separerar data i klasser i ett högdimensionellt utrymme. SVM fokuserar på att maximera marginalen mellan datapunkter och beslutsgränsen, vilket förbättrar dess generalisering. Det är särskilt effektivt i högdimensionella utrymmen och används ofta i olika applikationer.

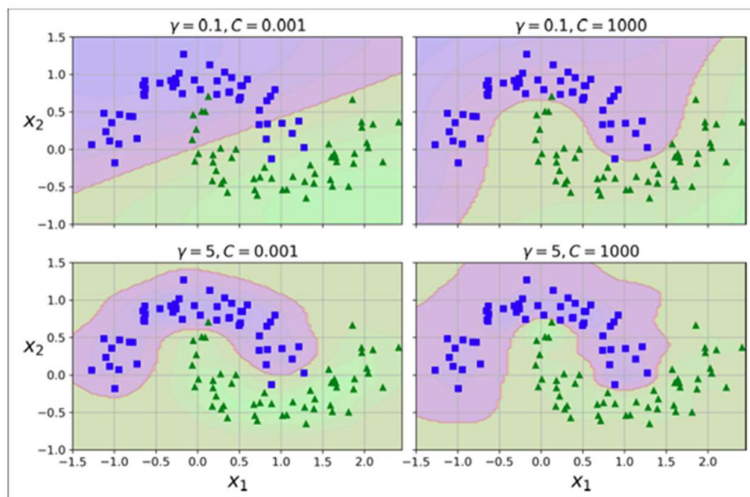


Figure 3. SVM classifiers using an RBF kernel

(Image taken from (p.161) in the second edition of Hands on machine Learning with Scikit-Learn Keras & Tensorflow, Aurélien Géron.)

With so many kernels to choose from, how can you decide which one to use? As a rule of thumb, you should always try the linear kernel first (remember that LinearSVC is much faster than SVC(kernel="linear")), especially if the training set is very large or if it has plenty of features. If the training set is not too large, you should also try the Gaussian RBF kernel; it works well in most cases. (Aurélien Géron, 2019)

2.3 Model Evaluation Framework

När vi arbetar med ett klassificeringsproblem kan vi utvärdera modellens prestanda med hjälp av mätvärden som Accuracy, Recall, F1-Score och Precision. I detta forskningsarbete kommer jag att förlita mig på Confusion Matrix och F1-Score som de primära måtten för att bedöma klassificerarnas prestanda. Confusion Matrix ger en detaljerad uppdelning av sanna positiva, sanna negativa, falska positiva och falska negativa, vilket ger insikter om modellfel. F1-poängen, som är det harmoniska medelvärde av Precision och Recall, är särskilt användbar för obalanserade datamängder och hyperparameterjustering, såsom grid search med cross-validation, har använts för att optimera klassificerare parametrar för bättre prestanda.

2.3.1 Confusion Matrix

Confusion matrix är en enkel tabell som visar hur väl en klassificeringsmodell presterar genom att jämföra dess prediktioner med de faktiska resultaten. Den delar upp förutsägelserna i fyra kategorier: korrekta prediktioner för båda klasserna (true positive and true negative) och felaktiga prediktioner (false positive and false negatives). Detta hjälper dig att förstå var modellen gör misstag, så att du kan förbättra den.

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Fig :3 Confusion matrix

2.3.2 HyperParameter

Jag använde GridSearchCV, som utför en uttömmande sökning över den angivna parametergrid. För exempel ställde jag justerade regulariseringsparametern C för att optimera balansen mellan underpassning och överanpassning för Logistic Regression. När det gäller SVM justerade jag C-värdet och kernel typ för att hitta den bästa modellkonfigurationen.

3. Metod

I det här kapitlet diskuteras ursprunget och beredningen av MNIST-datauppsättningen, inklusive databearbetningssteg som skalning och dimensionsreduktion med PCA.

3.1 Data Insamling

MNIST-datauppsättningen som erhålls med funktionen `fetch_openml` från `scikit-learn`-biblioteket, vilket säkerställer tillgänglighet och reproducerbarhet för data. Efter att ha hämtat data tillämpades följande förbearbetningssteg.

3.1.1 Feature Scaling

Funktionsvärdena normaliserades med hjälp av `StandardScaler` för att ta bort medelvärdet och skala dem till enhetsvariansen. Det här steget säkerställer att features är i samma skala, vilket är viktigt för modeller som är känsliga för funktionsstorlekar.

3.1.2 Dimensionality Reduction

Principal Component Analysis (PCA) tillämpades för att minska dimensionaliteten hos data samtidigt som 95 % av variansen bibehölls. Detta minskade beräkningsbelastningen och bidrog till att förbättra effektiviteten och generaliseringen av modellerna genom att eliminera överflödiga och mindre viktiga features.

3.1.3 Model Training

Den förberedda datan delades sedan upp i träning och testset, som fungerade som grund för träning och utvärdering av modellens algoritmer (Logistic Regression, Random Forest and Support Vector Machine). Efter att ha delat upp data tränades Logistic Regression med hjälp av grid search med korsvalidering för att optimera C-parametern, Random Forest tränades med 100 träd och varierande träddjup för att fånga komplexa mönster och Support Vector Machine (SVM) använde grid search för att tune C och RBF-kernel för icke-linjära separationer. Alla modeller använde PCA-transformerade data för effektivitet och minskad överanpassning.

3.1.4 Model Evaluation

Modellens prestanda utvärderades med hjälp av accuracy, precision, recall och F1-score, med en klassificeringsrapport för detaljerade insikter. Confusion matrix för alla modeller framhäver klassificeringar. Korsvalidering under grid search säkerställde robust hyperparameterval.

4. Resultat och Performance

Models	Accuracy	F1-Score
SVM	0.96443	0.96
Random Forest	0.93679	0.94
Logistic Regression	0.92186	0.92

Tabell 1: Accuracy and F1-score of different models

4.1 Support Vector Machine (SVM)

SVM-modellen (Support Vector Machine) uppvisade enastående prestanda och uppnådde den högsta accuracy på 0,96443 bland alla testade modeller. Detta tyder på att SVM var mycket effektiv när det gällde att fånga de komplexa mönstren i MNIST-datauppsättningen, särskilt på grund av användningen av kernel: rbf, som är väl lämpad för att hantera icke-linjära relationer. Precision, Recall och F1-poängen framhäver sannolikt dess förmåga att klassificera siffrorna korrekt med minimal felklassificering.

4.2 Logistic Regression

Logistic Regression fungerade som en solid baslinjemodell och uppnådde en accuracy på 0,92186. Även om den presterade ganska bra, var dess accuracy lägre jämfört med Random Forest och SVM, vilket kan tillskrivas dess linjära karaktär. Logistic Regression är en enklare linjär modell, så dess noggrannhet kan begränsas av komplexiteten hos MNIST-datauppsättningen.

4.3 Random Forest

Random Forest visade stark prestanda med en accuracy på 0,93679, vilket överträffade Logistic Regression. Som en ensemblemetod kunde den effektivt fånga mönster i data genom att utnyttja flera beslutsträd, vilket minskade sannolikheten för överfitting. Det föll dock under SVM:s prestanda, potentiellt på grund av begränsningar i att hantera mer intrikata icke-linjära mönster i datamängden.

5. Slutsats

Observationerna visar att SVM utmärker sig i accuracy och prediktiv kraft, Random Forest ger ett starkt alternativ med ensembleinlärning, och Logistic Regression är en effektiv, om än mindre kraftfull, baslinjemodell.

Vår första slutsats är att vår studie ger ett tydligt svar på vår första forskningsfråga: Vilka förbearbetningssteg och feature extraktionsmetoder förbättrar prestandan för sifferigenkänning? Det finns många förbearbetningssteg och funktionsextraktionsmetoder förbättrar prestandan för sifferigenkänning. I min modell använder jag några av stegen till exempel: Normalisera data (med StandardScaler), hantera saknade värden (imputation) och tillämpa dimensionsreducerande tekniker som PCA för att ta bort noise och minska komplexiteten.

Utöver vår första slutsats kommer vi nu att ta upp den andra forskningsfrågan: Vilken maskininlärningsmodell (Logistisk regression, Random Forest eller SVM) kommer att ge mer än 90% accuracy? Alla tre modellerna kan uppnå över 90% accuracy med lämplig preprocessing och justering av hyperparametrar. SVM presterar dock i allmänhet bäst, vilket framgår av resultaten, med en accuracy på 0,96443, följt av Random Forest och logistisk regression.

Det är dock viktigt att erkänna vissa begränsningar i denna studie, såsom den potentiella effekten av olika datauppsättningar och omfattningen av de förbehandlingsmetoder som utforskas. Framtida forskning kan utforska ytterligare datauppsättningar, avancerade förbehandlingstekniker och ensemblemetoder för att ytterligare förfinas modellens prestanda.

Part 2. Teoretiska frågor

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

Svar: Kalle delar upp sin data i "Training", "Validation" och "Test" set, varje del tjänar ett distinkt syfte i maskininlärningsarbetsflödet och det säkerställer att modellen tränas och utvärderas ordentligt.

Träning: Modellen använder träningsuppsättningen för att förstå mönster, relationer och egenskaper. Detta är den faktiska datamängden från vilken en modell train.

Validering: Valideringssetet används för att finjustera modellen under träning. Det hjälper till att utvärdera modellens prestanda och justera hyperparametrar utan att överfitta träningssetet.

Test: Detta är data som modellen aldrig har sett förut och används endast efter avslutad träning. Det ger en unbiased evaluation av modellens prestanda i verkliga scenarier.

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"?

Svar: Julia kan använda cross validation på träningsdata för att jämföra modellprestanda. Genom att dela upp träningsdata i flera delmängder kan hon träna och testa varje modell över dessa uppdelningar. Beräkna sedan prestandan genom att använda root mean squared error eller mean squared error. Julia väljer modellen med den bästa genomsnittliga prestandan över folds, och säkerställer att den generaliserar väl till osynliga data. Denna metod hjälper i avsaknad av en dedikerad valideringsdatauppsättning.

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

Svar: Regressionsproblem är när en beroende variabel förutsägs som ett kontinuerligt numeriskt värde baserat på en eller flera oberoende egenskaper. Den hittar samband mellan variabler så att förutsägelser kan vara korrekt eller göras.

Exempel på modell där vi använder regressionsproblem: Linjär regression, Lasso-regression, Random forest regressor etc.

Potentiella tillämpningsområden:

Regression är en statistisk metod som används inom finans, till exempel: att uppskatta huspriser baserat på faktorer som storlek, läge och antal sovrum med modeller som Linjär Regression eller Random Forest. Regressionsmodeller används också ofta som statistiska bevis för påståenden om vardagliga fakta.

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

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

Svar: Root Mean Squared Error (RMSE) is squared root of mean squared error (MSE) som är ett mått för att utvärdera modeller som görs på regressionsproblem. Det betyder mäter den genomsnittliga storleken av fel mellan predikt och actual värden i en regressionsmodell. Lägre RMSE bättre modellprestanda och bra prediktion.

n = Number of observations

y_i = Actual value

\hat{y}_i = Predicted value

RMSE används inom många områden, såsom väderprognoser och prediktion om huspriser. Den utvärderar precisionen hos modeller och används ofta för att jämföra och bedöma regressionsmodeller.

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

Svar: Ett klassificeringsproblem förutsäger en kategori eller class label (t.ex. spam kontra inte spam, man eller kvinna på bilden etc) för en given indata. Modeller som Logistic Regression, Random Forest och Support Vector Machine (SVM) används.

Tillämpningar inkluderar bestämning av handskriftssiffra, sjukdomsdiagnos, fraud detection och skräppostfiltrering etc.

Confusion Matrix.: En Confusion Matrix hjälper oss att se hur väl en modell fungerar genom att visa korrekta och felaktiga prediktioner. Den delar upp prediktionerna i fyra kategorier: korrekt prediktioner för båda klasserna (true positive och true negative) och felaktiga prediktioner (false positive och false negative). Detta hjälper oss att förstå var modellen gör misstag, så att vi kan förbättra den.

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

Svar: K-means klustring är en unsupervised maskininlärningsalgoritm som grupperar den unlabeled datamängden i olika kluster. Clustering hjälper oss att förstå vår data på ett unikt sätt genom att gruppera saker på deras likheter.

K-Means-klustring används i en mängd verkliga exempel eller affärsfall, till exempel: en återförsäljare kan analysera kundbeteende (t.ex. köphistorik, utgiftsvanor, demografi) och gruppera dem i kluster som budgetmedvetna köpare, frekventa kunder och premiumkunder för att skapa riktade marknadsföringsstrategier.

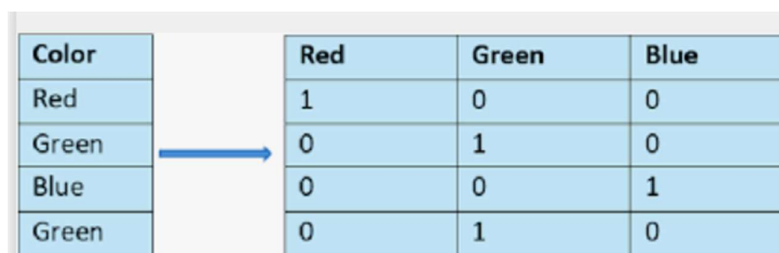
7. Förklara (gärna med ett exempel): Ordinal encoding, one-hot encoding, dummy variable encoding. Se mappen "l8" på GitHub om du behöver repetition.

Svar: Dessa tekniker väljs baserat på typen av kategoriska data och den maskininlärningsmodell som används.

Ordinal encoding: Ordinal encoding är en förbearbetningsteknik som används för att konvertera kategoriska data till numeriska värden som bevarar sin inneboende ordning.

Till exempel köper jag en skjorta och skjortorna är olika stora som small, medium och large. Så i det här fallet kan vi koda dem som små = 1, medelstora = 2, stora = 3. Detta förutsätter en naturlig ordning bland kategorierna.

One hot encoding: Den skapar binära (0/1) kolumner för varje kategori utan att anta någon ordning. Varje kategori representeras av en separat kolumn där en 1 anger dess närvaro, svar 0 för dess frånvaro. Till exempel har vi en datauppsättning med färgkolumn med kategorier och med hjälp av One hot encoding kan vi omvandla dessa kategoriska värden till numerisk form. I figur, färgen röd har ett värde på 1 medan andra färgkolumner innehåller 0.



Color	Red	Green	Blue
Red	1	0	0
Green	0	1	0
Blue	0	0	1
Green	0	1	0

Fig: One hot coding

Dummy Variable: Det är som one hot coding men det representerar C-kategorier med C1 binära variabler. Det bör användas om du använder en linjär regressionsmodell.

Till exempel har vi kategorivariabel Bil och med kategorierna Sedan, SUV, Lastbil. Om dummyvariabeln för SUV och lastbil båda är 0, är biltypen Sedan. Om dummyvariabeln för SUV är 1 när bilen är en SUV och annars 0. På samma sätt tar dummyvariabeln för lastbil värdet 1 för lastbilar medan SUV förblir 0.

Car Type	SUV	Truck
Sedan	0	0
SUV	1	0
Truck	0	1

Fig: Dummy variable

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?

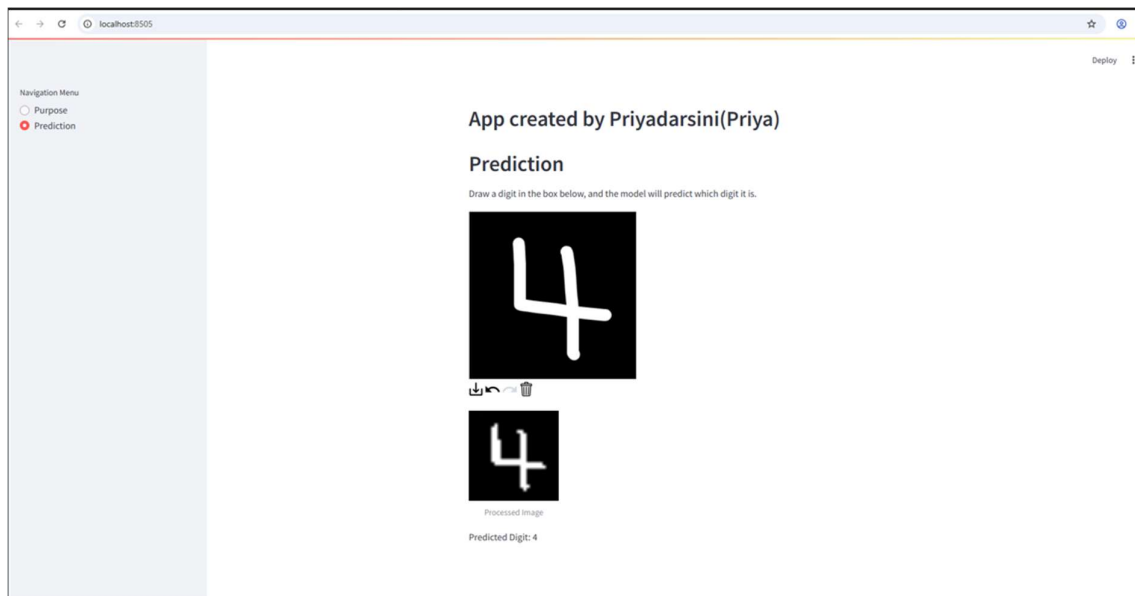
Svar: Både Göran och Julia har giltiga poäng. Göran identifierar korrekt distinktionen mellan "ordinal" och "nominal" datatyper, där ordningsdata har en inneboende ordning eller rangordning, medan nominella data saknar denna ordning. Julia, å andra sidan, lyfter fram betydelsen av tolkning, och antyder att sammanhang kan påverka hur data uppfattas. Till exempel, medan färger som rött och grönt vanligtvis är nominal, i ett specifikt scenario, som att bestämma attraktionskraft på en fest och som kan leda till en ordinal tolkning, tilldela en rangordning baserat på situationen.

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?

Svar: Streamlit är ett ramverk med öppen källkod för att skapa dataapplikationer i Python för maskininlärning och datavetenskapsteam. Det förenklar skapandet av interaktiva dashboards och visualiseringar (t.ex. plots). Streamlit kan användas för att bygga appar som visar upp maskininlärningsmodeller där användare kan mata in data och se prediktions eller resultat.

Till exempel: Jag har byggt en webbapplikation med Streamlit för att känna igen handskrivna siffror. För att skapa denna applikation och jag har använt den logistiska regressionsmodellen.



2 Självutvärdering

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

Att uppnå högre accuracy samtidigt som man förbereder en maskininlärningsmodell med hjälp av MNIST-datauppsättning kan vara en givande men ändå utmanande process. Några vanliga utmaningar jag stöter på är att välja optimala hyperparametrar och lösningen var att experimentera systematiskt med grid search och tillämpa PCA för att minska noise och redundans i funktioner.

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

Jag strävar efter att prestera mitt bästa i mitt arbete, och lägger kraft på varje steg för att säkerställa kvalitet. Men det är ditt perspektiv och betyg som verkligen kommer att spegla hur väl jag har lyckats.

3. Något du vill lyfta fram till Antonio?

Nej.

Appendix A

https://github.com/ppriya23/Machine_Learning

Källförteckning

Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems av *Aurelien Geron (2019).(Second Edition)*

<https://www.geeksforgeeks.org/machine-learning/?ref=shm>