Maskininlärning

Sifferigenkänning

Alia Atawna

EC Utbildning

Kunskapskontroll 2- Maskininlärning

2024-03

Abstract

This study delved into using machine learning to classify MNIST dataset's handwritten digits, aiming for over 90% accuracy. Through rigorous evaluation and optimization, the Support Vector Classifier (SVC) with specific hyperparameters {'C': 0.5, 'gamma': 1, 'kernel': 'poly'} stood out, achieving a test accuracy of 96.25%. This accomplishment not only surpassed our accuracy goal but also showcased the significant potential of machine learning in digit recognition tasks.

Innehållsförteckning

1 Inledning

Maskininlärning förändrar hur uppgifter hanteras som kräver mänsklig intelligens. I detta projekt undersöks hur maskininlärning kan användas för att lösa problemet att känna igen handskrivna siffror. Det är ett spännande exempel på hur tekniken kan hjälpa att tolka och förstå handskrift automatiskt.

Syftet med denna rapport är att utforska och jämföra olika maskininlärningsmodeller för klassificering av handskrivna siffror från MNIST-databasen, för att uppfylla syftet så kommer följande frågeställning att besvaras:

1. Är det möjligt att utveckla en maskininlärningsmodell som uppnår minst 90% noggrannhet i att klassificera handskrivna siffror från MNIST-datasetet?

2 Teori

2.1 Support Vector Classifier

Support Vector Classifier (SVC) är en teknik inom maskininlärning för att skilja mellan klasser genom att hitta det mest optimala avståndet mellan dem. Detta uppnås genom att använda supportvektorer och en kärnfunktion för att projicera data till ett högre dimensionellt utrymme där det blir enklare att dra en skiljelinje mellan klasserna. Hyperparametern C används för att balansera mellan att få en enkel modell och att klassificera träningsdatan korrekt. Kernelparametern bestämmer vilken typ av kärnfunktion som ska användas, medan gamma inom 'rbf'-kärnan reglerar hur mycket enskilda datapunkter påverkar klassificeringen. (Géron, 2019)

2.2 Random Forest Classifier

RandomForest Classifier är en effektiv maskininlärningsalgoritm som använder flera beslutsträd för att göra mer exakta förutsägelser och förbättra stabiliteten, genom att undvika överanpassning. Detta uppnås genom att skapa olika träd baserade på slumpmässiga urval av träningsdata och sedan kombinera deras beslut. Några viktiga inställningar i algoritmen inkluderar n_estimators, som är antalet träd i skogen, och max_depth, som begränsar hur djupt träden får vara. Det finns också parametrar som min_samples_split och min_samples_leaf, som hjälper till att kontrollera storleken på trädets noder och därmed modellens komplexitet. Justering av dessa parametrar kan hjälpa till att finjustera modellens prestanda och undvika överanpassning. (Géron, 2019)

2.3 Logistisk regression

Logistisk regression är en populär metod för att lösa binära klassificeringsproblem i maskininlärning, där målet är att förutsäga sannolikheten för att en observation tillhör en viss klass. Det använder en logistisk (sigmoid) funktion för att omvandla linjära uttryck till värden mellan 0 och 1, representerande sannolikheten för klassmedlemskap. Nyckelparametrar inkluderar C, som kontrollerar regleringens styrka för att undvika överanpassning, och solver, som bestämmer den algoritm som optimerar modellen. Logistisk regression är värdefull för dess förmåga att ge insikter om sannolikheten bakom prediktionerna, vilket är användbart i många tillämpningar som medicinsk diagnos och kundsegmentering. (Géron, 2019)

3 Metod

3.1 Dataförberedelse

Projektet inleddes med att ladda ned MNIST-databasen via fetch_openml funktionen från sklearn.datasets. Denna dataset består av 70 000 handskrivna siffror representerade i 784 dimensioner (28x28 pixlar). För att minska träningskostnaderna och tiden, användes en delmängd av datasetet där de första 12 000 exemplaren delades in i tränings-, validerings- och testuppsättningar. Data normaliserades och standardiserades med hjälp av StandardScaler för att förbättra modellernas prestanda.

3.2 Modellval och träning

Tre olika typer av maskininlärningsmodeller valdes för projektet baserat på deras förmåga att hantera högdimensionella data och klassificeringsuppgifter:

- Support Vector Classifier (SVC): En SVC-modell tränades med en rad olika hyperparametrar (C, kernel, gamma) för att hitta den bästa konfigurationen med GridSearchCV.
- Random Forest Classifier: Användes för att utnyttja ensemble lärande för bättre generalisering och robusthet mot överanpassning. Hyperparametrar som antalet träd och trädets maximala djup optimerades också med GridSearchCV.
- Logistisk Regression: Trots dess enkelhet, testades logistisk regression för dess förmåga att utföra binär klassificering i multiklass-scenarion genom one-vs-rest strategi.

För varje modell utfördes hyperparameteroptimering för att balansera modellkomplexitet med träffsäkerhet på valideringsdatan.

3.3 Utvärdering

Modellernas prestanda utvärderades med hjälp av valideringsdatan för att undvika överanpassning och säkerställa att modellerna hade generell förmåga. Utvärderingsmetrikerna inkluderade noggrannhet och en detaljerad analys med hjälp av confusion matrix. SVC-modellen med en polynomiell kärna visade sig ha högst valideringsnoggrannhet och valdes därför för ytterligare utvärdering på testdatasetet.

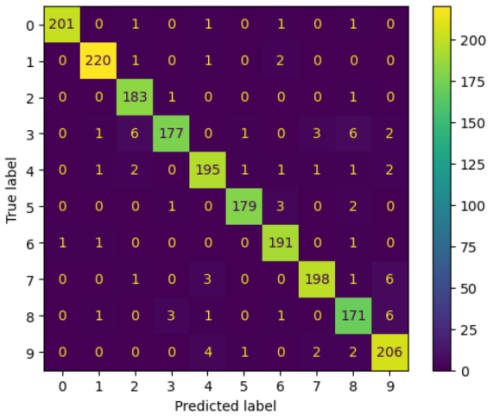
4 Resultat och Diskussion

4.1 Modellval

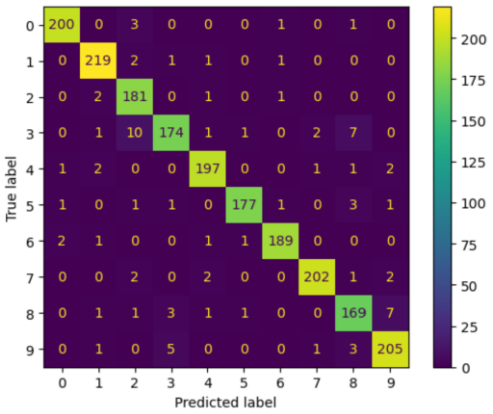
I processen för att välja de bästa modellerna för klassificering av handskrivna siffror från MNIST-datasetet, användes GridSearchCV för att finjustera hyperparametrarna för både Support Vector Classifier (SVC) och Random Forest Classifier. Optimeringen resulterade i att SVC med hyperparametrarna {'C': 0.5, 'gamma': 1, 'kernel': 'poly'} uppnådde den högsta valideringsnoggrannheten på 96.06%. För Random Forest Classifier var de optimala hyperparametrarna {'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}, vilket ledde till en valideringsnoggrannhet på 95.65%. Nedan följer en sammanfattande tabell över modellernas prestation samt confusion matrix för djupare insikter i deras klassificeringsförmåga.

Accuracy för valda modeller	
SVC	96,05 %
Random Forest Classifier	95,65 %
Logistisk regression	91,60 %

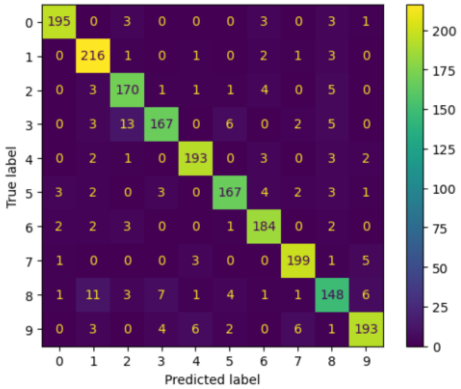
Tabell 1: Accuracy för valda modeller



Figur 1: Confusion Matrix SVC.



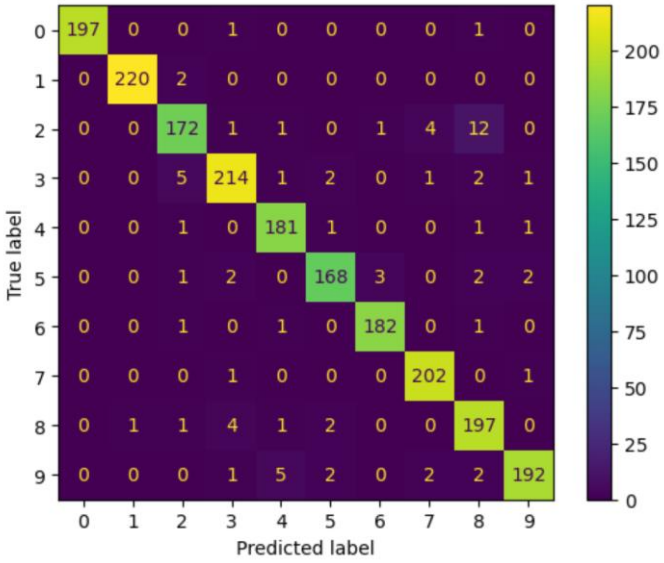
Figur 2: Confusion Matrix Random Forest Classifier.



Figur 3: Confusion Matrix Logistisk regression.

Dessa resultat och den efterföljande analysen av confusion matrix bidrog till valet av den mest lämpliga modellen för uppgiften, baserat på både noggrannhet och förmågan att skilja mellan de olika siffrorna i MNIST-datasetet.

Efter jämförelse och optimering av hyperparametrar för Support Vector Classifier (SVC) och Random Forest Classifier, framträdde SVC som den överlägsna modellen. Optimeringen genom GridSearchCV avslöjade att SVC med inställningarna {'C': 0.5, 'gamma': 1, 'kernel': 'poly'} presterade bäst, med en testnoggrannhet på 96.25%. Denna höga noggrannhet understryker SVC-modellens förmåga att effektivt klassificera komplexa handskrivna siffror, vilket gör den till ett robust verktyg för sifferigenkänning. Den optimala konfigurationen, med en polynomiell kärna, visade sig vara avgörande för modellens framgång, tack vare dess förmåga att hantera datasetets icke linjära egenskaper.



Figur 4: Confusion SVC.

Confusion matrix för SVC gav ytterligare insikter i modellens prestanda och hjälpte till att identifiera områden för eventuell förbättring genom att avslöja specifika mönster i felklassificeringarna. Med dessa resultat valdes SVC som den slutgiltiga modellen för projektet. Dess framgång öppnar för möjligheter till framtida utforskningar, inklusive finjusteringar, tester på andra dataset, och utvecklingen av praktiska applikationer inom automatisk sifferigenkänning.

5 Slutsatser

I detta projekt utforskades möjligheten att utveckla en maskininlärningsmodell som kan klassificera handskrivna siffror från MNIST-datasetet med en noggrannhet på minst 90%. Genom att noggrant jämföra och optimera olika modeller, framkom Support Vector Classifier (SVC) med hyperparametrarna {'C': 0.5, 'gamma': 1, 'kernel': 'poly'} som den mest effektiva, uppnående en testnoggrannhet på 96.25%. Detta resultat bekräftar att det inte bara är möjligt att nå projektets mål, utan också att överstiga det avsevärt med rätt modell och parametrar.

Resultaten från detta arbete bekräftar därmed att svar på den initiala frågeställningen är ja, det är definitivt möjligt att utveckla en maskininlärningsmodell som uppnår och överstiger en noggrannhet på 90% för klassificering av handskrivna siffror i MNIST-datasetet. Denna framgång understryker potentialen hos maskininlärning för att effektivt tolka och förstå handskrift, vilket öppnar upp för framtida applikationer och ytterligare forskning inom området.

6 Teoretiska frågor

1. **Kalle delar upp sin data i ”Träning”, ”Validering” och ”Test”, vad används respektive del för?**
   Träning: Används för att träna vår modell, dvs lära upp modellen. (Wikipedia contributors, n.d.)
   Validering: Används för att utvärdera modellerna samtidigt som modellens hyperparametrar justeras. Den bästa modellen väljs ut. (Wikipedia contributors, n.d.)
   Test: Används för att testa den valda modellen, det används när en modell är helt tränad. (Wikipedia contributors, n.d.)

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 ”valideringsdataset”?**
   För att välja mellan modeller utan ett valideringsdataset, kan Julia använda korsvalideringstekniker som K-Fold korsvalidering. Denna metod delar hennes träningsdata i "k" antal delar, tränar modellen på k-1 av dessa delar och validerar prestandan på den kvarvarande delen. Denna process upprepas "k" gånger, med en annan del som valideringsset varje gång. Genom att jämföra resultaten från dessa iterationer kan hon avgöra vilken modell som generellt presterar bäst. (Brownlee, 2019)

3. **Vad är ”regressionsproblem? Kan du ge några exempel på modeller som används och potentiella tillämpningsområden?**
   Ett regressionsproblem syftar till att prediktera ett kontinuerligt värde baserat på en eller flera oberoende variabler. Exempel på modeller som används för regressionsproblem inkluderar Linjär regression, där relationen mellan variablerna modelleras som en linje, och Random Forest Regressor, en ensemblemetod som använder flera beslutsträd. Tillämpningsområden kan vara allt från fastighetsprisförutsägelser, där man utifrån egenskaper som storlek och läge förutsäger bostadskostnad, till aktieprisanalys, där man försöker förutsäga framtida aktiepriser baserat på historiska data. (MachineLearningMastery.com, 2019)

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

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

   RMSE står för Root Mean Squared Error, det är ett mått på hur väl en modell kan förutsäga resultat i förhållande till de faktiska värdena. Den beräknas som kvadratroten av medelvärdet av kvadraterna av skillnaderna mellan de förutsagda värdena och de faktiska värdena.
   Man tar skillnaden mellan en prediktion och respektive observerat värde: $(\boldsymbol{y_i} - \boldsymbol{\hat{y}_i})$ kallas för Error. Det spelar ingen roll om det är en positiv eller negativ skillnad, därför kvadreras: $(y_i - \hat{y}_i)^2$ kallas för Squared Error.
   Medelvärdet för Squared Error räknas ut, summan av alla Squared Error divideras med antal observationer: $\frac{1}{n}\sum_{i}(\boldsymbol{y_i} - \boldsymbol{\hat{y}_i})^{\boldsymbol{2}}$ kallas för Mean Squared Error.
   Roten ur Mean Squared Error, så siffran är på samma skala som datan och därmed lättare att tolka: $\sqrt{\frac{1}{n}\sum_{i}(\boldsymbol{y_i} - \boldsymbol{\hat{y}_i})^{\boldsymbol{2}}}$.

RMSE är användbart för att jämföra passformen mellan olika regressionsmodeller, den modell med lägsta RMSE-värde anses passa bäst. (Statology, 2021).

5. **Vad är ”klassificieringsproblem? Kan du ge några exempel på modeller som används och potentiella tillämpningsområden? Vad är en ”Confusion Matrix”? ’**
   Ett klassificieringsproblem handlar om att prediktera en klass/diskret värde för en given observation. Exempel på modeller för detta inkluderar Logistisk Regression och Support Vector Machines. Dessa modeller kan tillämpas på många områden, som e-postfiltrering (spam eller inte), medicinsk diagnos, och bildigenkänning. En "Confusion Matrix" är ett verktyg för att summera prestandan av en klassificeringsalgoritm genom att visa antalet korrekta och felaktiga förutsägelser, uppdelat per klass, vilket gör det möjligt att detaljerat utvärdera modellens prestanda. (Brownlee, 2020)

6. **Vad är K-means modellen för något? Ge ett exempel på vad det kan tillämpas på.**
   K-means är en osuperviserad inlärningsalgoritm som delar in data i k cluster baserat på likheter mellan datapunkter. Den används för exempelvis segmentering av kundbeteenden och bildkomprimering. Algoritmen kräver att antalet cluster specificeras i förväg och kan vara känslig för valet av initiala centrum samt för outliers. (Simplilearn, 2023).

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 tilldelar varje unik kategori ett heltalsvärde, användbart för kategoriska variabler med en naturlig rangordning. One-hot encoding omvandlar kategoriska variabler utan naturlig ordning till binära kolumner, där varje kategori representeras av en kolumn. Dummy variable encoding liknar one-hot encoding men använder en binär kolumn mindre för att undvika redundans och potentiell multicollinearitet i vissa modeller. Dessa tekniker är viktiga för att omvandla kategoriska data till ett numeriskt format som maskininlärningsmodeller kan arbeta med. (MachineLearningMastery, 2020).

8. **Göran påstår att datan antingen är ”ordinal” eller ”nominal”. Julia säger att detta måste tolkas. Hon ger ett exempel med att färger såsom {röd, grön, blå} generellt sett inte har någon inbördes ordning (nominal) men om du har en röd skjorta så är du vackrast på festen (ordinal) – vem har rätt?**
   Julia har rätt i att kategoriseringen av data som "ordinal" eller "nominal" kan bero på kontexten och hur datan tolkas. Medan färger generellt sett anses vara nominala data eftersom de inte har en naturlig ordning, kan de i vissa sammanhang tolkas som ordinala beroende på ett specifikt kriterium eller en situation, som Julias exempel med färgen på en skjorta som påverkar vackrast på en fest. Detta illustrerar vikten av att överväga kontext när man bestämmer datatyp.

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 öppen källkodsappbibliotek som gör det enkelt för utvecklare att skapa och dela datadrivna webbapplikationer skrivna helt i Python. Det används ofta för att snabbt utveckla datavisualiseringar, interaktiva dashboards, och maskininlärningsprototyper utan att behöva oroa sig för frontend utveckling. Det erbjuder ett enkelt gränssnitt för att integrera olika datavisualiseringsbibliotek och hantera användarinput.

7 Självutvärdering

1. **Utmaningar du haft under arbetet samt hur du hanterat dem.**
   En av de svårigheter jag stötte på var att hitta en startpunkt för arbetet. Att veta vilket steg som skulle tas först var inte enkelt. Men när jag väl började experimentera och utforska olika tillvägagångssätt blev det gradvis lättare att fortsätta uppgiften.

2. **Vilket betyg du anser att du skall ha och varför.**
   Jag anser att betyget G är passande för mig, eftersom jag tror att jag uppfyllt de uppsatta kriterierna för detta betyg.

3. **Något du vill lyfta fram till Antonio?**
   Tacksam för all hjälp vi fått.

Källförteckning

Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (2nd ed.). O'Reilly Media. Hämtad 16 mars, 2024, från http://14.139.161.31/OddSem-0822-1122/Hands-On_Machine_Learning_with_Scikit-Learn-Keras-and-TensorFlow-2nd-Edition-Aurelien-Geron.pdf

Wikipedia contributors. (n.d.). Training, validation, and test data sets. Wikipedia. Hämtad 8 mars, 2024, från https://en.wikipedia.org/wiki/Training,_validation,_and_test_data_sets

Brownlee, J. (2019). A Gentle Introduction to Model Selection for Machine Learning. Machine Learning Mastery. Hämtad 8 mars, 2024 från https://machinelearningmastery.com/a-gentle-introduction-to-model-selection-for-machine-learning/

Brownlee, J. (2019). Difference Between Classification and Regression in Machine Learning. Machine Learning Mastery. Hämtad 8 mars, 2024 från https://machinelearningmastery.com/classification-versus-regression-in-machine-learning/

Zach. (2021). How to Interpret Root Mean Square Error (RMSE). Statology. Hämtad 8 mars, 2024 från https://www.statology.org/how-to-interpret-rmse/

Brownlee, J. (2020, August 15). What is a Confusion Matrix in Machine Learning. MachineLearningMastery. Hämtad 8 mars, 2024 från https://machinelearningmastery.com/confusion-matrix-machine-learning/

Simplilearn. (2023). K-means Clustering Algorithm: Applications, Types, and Demos. Hämtad 8 mars, 2024 från https://www.simplilearn.com/tutorials/machine-learning-tutorial/k-means-clustering-algorithm

Brownlee, J. (2020, August 17). Ordinal and One-Hot Encodings for Categorical Data. Machine Learning Mastery. Hämtad 8 mars, 2024 från https://machinelearningmastery.com/one-hot-encoding-for-categorical-data/.