SupportVectorClassifier

vs.

RandomForestClassifier

Jämförelse med hjälp av MNIST dataset

Marcus Eklund

EC Utbildning

Kunskapskontroll – Machine Learning

2024-03

Abstract

After comparing Support Vector Classifier and Random Forest Classifier, it's found that the Random Forest Classifier with GridSearchCV achieves the highest accuracy of 0.94, surpassing other models. However models requiring parameter adjustments do not achieve a significant increase in accuracy compared to the time it takes to adjust the parameters and are therefore discarded. Despite slight overfitting observed during testing, Random Forest Classifier with it’s default parameters performs satisfactorily, aligning with expectations. However, it's worth considering exploring additional models beyond these, as some may offer superior performance for the MNIST dataset. Nevertheless, such exploration lies outside the scope of this study.

Innehållsförteckning

1 Inledning

I vår allt mer digitaliserade värld så finns det ett enormt behov av effektiv datahantering. Ett sådant behov är speciellt tydligt när det kommer till att avläsa handskriven text och siffror. Vi har fortfarande stor använding för handskrift men har inte tid att läsa av det manuellt och behöver därför lära datorer hur den kan göra det åt oss. När det kommer till att läsa av siffror så är det väldigt användbart för t.ex. posthantering där en maskin läser av postkoden för att kunna sortera den till rätt terminal. För att testa och utvärdera modeller för detta syfte så finns det ett dataset som är otroligt användbart, MNIST. MNIST innehåller ett stort antal handskrivna siffror i form av bilder på 28x28 pixlar.

1.1 Syfte och frågeställning

I denna uppsats så jämför vi 2 olika maskininlärningsmodeller för avläsning av handskrivna siffror på MNIST-datasetet: Support Vector Classifier (SVC) och Random Forest Classifier. Genom att utvärdera deras prestanda i avseende på avläsning och klassificering av handskrivna siffror, strävar vi efter att identifiera deras styrkor och svagheter.

För att ta reda på detta så kommer följande frågeställningar besvaras:

1. Vilken modell är säkrast i sin klassificering?
2. Vilken är snabbast?
3. Vilken modell borde vi använda?

2 Teori

I denna del går vi igenom relevant teori om datasetet och modellerna som används för att förstå projektet.

2.1 MNIST

MNIST datasetet (Modified National Institute of Standards and Technology database) är en stor samling av handskrivna siffror. Den har totalt 70 000 exemplar varav 60 000 är träningsset och 10 000 är testset. Den är en del av ett större testset, NIST Special Database 3 och Special Database 1 som är siffror skrivna av anställda på United States Census Bureau och amerikanska gymnasieelever respektive. Exemplen i dessa databaser är 20x20 pixlar bilder vars pixlar är enbart svarta eller vita. I MNIST så har dem placerats i bilder med 28x28 pixlar och har genomgått en bildhanteringsprocess som gett pixlarna en gråskala istället för svart eller vit (*LeCun, Y.*). Det innebär att varje pixel kan ha ett gråskalevärde från 0 för svart till 255 för vit (*Bovik, A. C.*). Där vi tidigare har en pixel som var helt svart bredvid en helt vit så har vi nu en mer gradiell övergång som ger mjukare kanter på linjerna i bilderna.

Bovik, A. C. (2005). Handbook of Image and Video Processing. (2. edition)

LeCun, Y. THE MNIST DATABASE of handwritten digits

2.2 Support Vector Classifier

Support Vector Classifier (SVC) är en så kallad ”supervised learning method” (övervakad inlärningsmetod) för att klassificera data inom maskininlärning. Övervakad innebär att den behöver märkt träningsdata för att kunna lära sig att göra förutsägelser. Alltså att vi redan vet vad för klassificeringar som finns och lär den att data med vissa attribut till hör en viss klass. SVC fungerar genom att skapa en modell som separerar datan i olika klasser med hjälp av en ”decision boundry” eller beslutsgräns, som försöker maximera avståndet mellan närliggande datapunkter av olika klasser. Detta gör det möjligt att klassificera ny oidentifierad data genom att placera den på ena eller andra sidan av gränsen.

Scikit-Learn 1.4. Support Vector Machines

2.3 Random Forest Classifier

Random Forest Classifier är en ensemble-lärande metod inom maskininlärning som används för klassificering av data. Ensemble-lärande innebär att den kombinerar flera olika modeller idetta fallet konstruerar den flera beslutsträd under träning och kombinerar deras resultat för att förbättra den övergripande prediktionskraften. Varje beslutsträd tränas på olika delmängder av träningsdata och använder olika slumpmässiga urval av funktioner vid varje nod för att bygga en variation av träd. Vid klassificering eller förutsägelse får varje träd en röst, och klassen som erhåller flest röster blir den slutgiltiga förutsägelsen.

Scikit-Learn. Sklearn.ensemble.RandomForestClassifier

IBM. What is random forest?

3 Metod

I denna del går vi igenom hur undersökningen genomfördes.

För undersökningen användes datasetet MNIST som hämtades genom Python paketet Scikit-learn.

Scikit-learn är också det som tillät oss att använda maskininlärningsmetoderna SVC och Random Forest Classifier samt metoder för att granska och bedöma hur modellerna presterar.

Jag började med att läsa om datasetet för att få en förståelse för vad det innehåller och hur det ser ut. Som nämnt i teoridelen så innehåller MNIST 70 000 bilder på hanskrivna siffror i 28x28 pixlar.

Bild 1 nedan är dem 3 första siffrorna i datasetet och när vi tittar på de motsvarande 3 första etiketterna så är det ”5, 0, 4” vilket verkar stämma med bilden.

Från detta kan vi dra slutsatsen att bilderna inte är ordnade och det hjälper oss med träningen då vi kan ta en mindre del av setet som representation för hela och träna modellerna på den vilket hjälper effektivisera träningen. Jag tog 7 000 bilder istället och delade upp det i ett träningsset, valideringsset och testset med 5 000, 1 000 och 1 000 respektive.

Jag började med att träna båda modellerna med deras ursprungliga parametrar. För att jämföra dem så använde jag accuracy_score vilket är en funktion i scikit-learn som ger poäng från 0 till 1 för hur väl en modell kan klassificera data.

Vidare så testade jag om jag kunde hitta andra parametrar som gjorde modellerna bättre så jag började med att använda GridSearch för SVC som jämför olika parametrar för att hitta dem som ger bäst totala resultat.

Jag testade även RandomizedSearch som också testar parametrar men där vi kan ge den sekvenser av värden som den väljer från slumpmässigt ett visst antal gånger som vi bestämmer. Jag använde GridSearch även för Random Forest Classifier med samma antal parametrar som för SVC.

Från dessa så valde jag Random Forest Classifier med ursprungliga parametrar och testade den mot vår testdata.

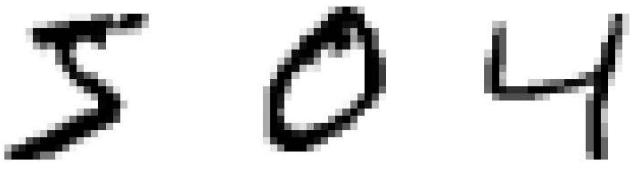

Bild 1: 3 första siffrorna i MNIST

4 Resultat

Out-of-Box	**Time**	**Accuracy**
Support Vector Classifier	7s	0.926
Random Forest Classifier	5.9s	0.939

Tabell 1: Out-of-Box test för SVC och Random Forest

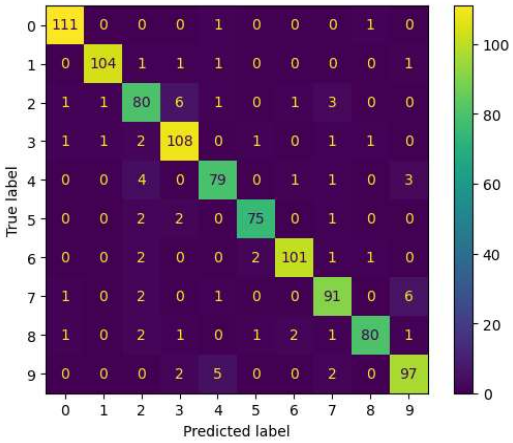

Bild 2: SVC confusion matrix

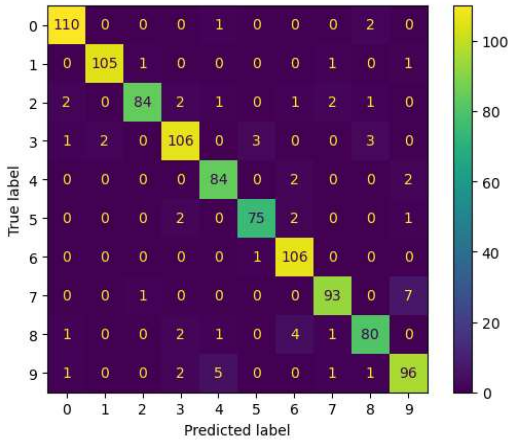

Bild 3: RandomForest confusion matrix

I de initiella testerna så presterade Random Forest Classifier bättre i både tid och säkerhet vilket kan ses i *Tabell 1*. Tiden är dock försumbar mellan dessa två då skillnaden enbart är 1,1 sekunder.

I *Bild 2* och *Bild 3* ser vi en Confusion Matrix som visar hur många gånger modellen predikterade en viss siffra mot den korrekta siffran. Diagonalen är där modellen predikterat korrekt siffra medans dem andra är när den predikterat fel. Vi ser att det skiljer sig väldigt lite mellan modellerna men att dem båda generelt sett har större problem med siffran 7 som båda modellerna, 6 gånger för SVC och 7 för Random Forest, trodde var siffran 9. SVC har även gissat fel 6 gånger för siffran 2 som den trodde var siffran 3 men Random Forest gjorde bara det felet 2 gånger.

I parametertesterna så ser vi i *Tabell 2* att Random Forest Classifier med GridSearch gav den högsta säkerheten på 0.94 men tog längst tid. Den snabbaste av dessa var Random Forest Classifier med RandomizedSearch på 3 minuter och 27 sekunder vilket var nästan 1 minut snabbare än den näst snabbaste med en säkerhet på 0.939.

När jag sedan testade Random Forest Classifier med ursprungliga parametrarna på testdatan så gav den en säkerhet på 0.937 och i *Bild 3* ser vi att den bettede sig nästan lika dant som den gjort på valideringsdatan men den hade det lite svårare med 7:orna som den 11 gånger predikterade som siffran 9.

Parameter tuning	Support Vector Classifier		Random Forest Classifier	
Parameter Search Function	**Time**	**Accuracy**	**Time**	**Accuracy**
GridSearchCV	4m47s	0.939	4m53s	0.94
RandomizedSearchCV	4m26s	0.932	3m27s	0.939

Tabell 2: Parameter tuning där GridSearchCV och RandomizedSearchCV används på SVC och Random Forest för att hitta bättre parametrar.

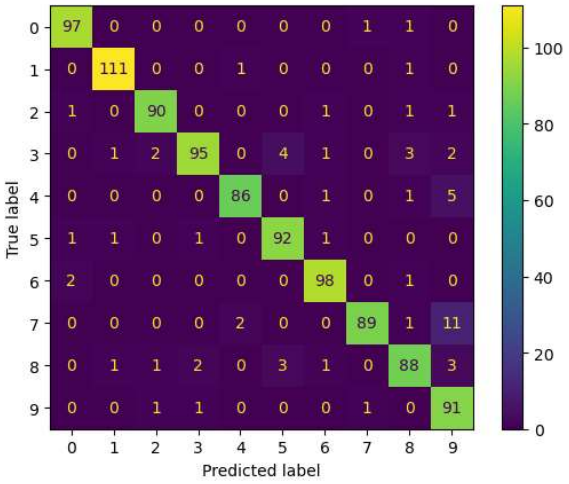

Bild 4: Confusion Matrix för Random Forest Classifier på testdatan.

5 Slutsatser

Nu kan vi svara på frågorna som vi ställde oss i början av undersökningen.

1. Vilken modell är säkrast i sin klassificering?
2. Vilken modell är snabbast?
3. Vilken modell borde vi använda?

När vi jämför alla modellerna så ser vi att Random Forest utan förändrade parametrar, Random Forest med RandomizedSearchCV och Support Vector Classifier låg på 0.939 i säkerhet men Random Forest Classifier med GridSearchCV gav den högsta säkerheten på 0.94. Alltså kan vi säga att från dessa så är Random Forest Classifier där GridSearchCV används den bästa modellen på att prediktera korrekt siffror från MNIST.

Tittar vi på tid så kan vi direkt utesluta modellerna som använder någon typ av parameter finjustering då dem inte kan mäta sig med modellerna som använde sina ursprungliga parametrar men frågan är då om deras säkerhet är bra nog för vad vi vill använda dem till. För det måste vi titta på säkerheten tillsammans med hastigheten.

Då jag inte kunde komma till en högre säkerhet än 0.94 och Random Forest Classifier redan från början uppnåde nästan samma säkerhet så anser jag att det inte är värt att spendera massa tid på att leta efter bättre parametrar. När modellen sedan testades på testdatan så fick vi resultatet 0.937 vilket tyder på att modellen är lite overfittad på datan men med en väldigt liten marginal. Över lag presterar den så som vi förväntat oss att den skulle.

Slutligen skulle jag säga att trots att modellen presterar väl så skulle det vara värt att titta på andra modeller utöver dessa också då det finns många som presterar bättre än dessa två för MNIST datasetet men det är utanför denna undersökning.

6 Teoretiska frågor

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

Träningsdata är den data som modellen lär sig på, validering är för att se hur väl den presterar så att vi finjustera och förbättra den. Test är den slutgiltiga testet för hur väl modellen presterar med ny osedd 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”?

Hon kan använda cross validation score och Root Mean Square Error(RMSE) som delar upp träningsdatan i mindre delar och testar dem mot varandra och ger oss sen ett värde.

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

Ett problem som har en beroende variabel som har kontinuerliga värden. Ett exempel är att lön är beronde på ålder och man kan då visa det med en linjär regression.

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

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

Root Mean Square Error tar roten ur medelvärdet av skillnaderna från sanna till predikterade värdena. Destå lägre värde destå bättre eftersom den tittar på hur fel modellen är men när vi skriver koden så byter vi tecken när vi tar roten ur för att scikit-learn räknar högre värde som bättre.

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

Klassificeringsproblem är där vi vill dela in datan i olika grupper(klasser). Ett exempel är som i denna rapporten, att avgöra vilken siffra som är skriven från handskrift och för att göra det kan vi använda Random Forest Classifier. En Confusion Matrix är ett verktyg för att se hur klassificeringsmodellen har lyckats och inte lyckats prediktera.

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

K-means är en unsupervised learning modell. Vi vet inte vad för samband som datan har utan låter modellen gruppera datan som kan sedan ge oss insikt i vad för samband som kan finnas, till exempel kunddata. Det kanske finns samband i ålder eller inkomst som påverkar hur kunden spenderar och vad dem köper.

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

Alla dessa används för att hantera kategorisk data då vi nästan alltid måste ha numerisk data för att kunna använda maskininlärning. Ordinal encoding är för kategorisk data som har någon slags ordningsrelation mellan kategorierna. Det enklaste exemplet är placeringar i en tävling, [”första”, ”andra”, ”tredje”]. One-hot encoding är för oordnade kategorier där vi inte vill att den ska dra samband/relationer mellan dem olika kategorierna, till exempel [”svart”, ”vit”, ”blå”]. Vi gör det genom att ge varje kategori ett binärt värde(0 eller 1). Dummy variable encoding fungerar som one-

hot men ger en mindre variabel. I tidigare exempel så är svart: [0, 0, 1], vit: [0, 1, 0] och blå: [1, 0, 0] men om vi vet att dem tre är unika och vi vet vad dem två första är så måste den sista vara blå och det betyder att vi inte behöver ge den en binär variabel. Dem blir då svart: [0, 1], vit: [1, 0] och blå: [0, 0]. Detta behövs till exempel för linjär regression.

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. Det kan finnas en ordning på datan som inte är direkt uppenbar från kategorinamnen.

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 för att bygga och dela appar för data och machine learning.

7 Självutvärdering

1. Utmaningar du haft under arbetet samt hur du hanterat dem.
   Att skriva en rapport på detta sättet tycker jag verkligen inte om och har svårt för men det hjälpte att prata igenom det med andra.

2. Vilket betyg du anser att du skall ha och varför.
   G. Jag anser att jag uppfyllt alla krav för G.

3. Något du vill lyfta fram till Antonio?

Källförteckning

Bovik, A. C. (2005). Handbook of Image and Video Processing. (2. edition)

Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow. (2. edition)

LeCun, Y. THE MNIST DATABASE of handwritten digits. Hämtad 3 april, 2024, från Yann LeCuns hemsida http://yann.lecun.com/exdb/mnist/

Scikit-Learn. 1.4. Support Vector Machines. Hämtad 3 april, 2024, från Scikit_Learns sida https://scikit-learn.org/stable/modules/svm.html

Scikit-Learn. Sklearn.ensemble.RandomForestClassifier. Hämtad 3 april, 2024, från Scikit-Learns sida https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

IBM. What is random forest? Hämtad 3 april, 2024, från IBMs sida https://www.ibm.com/topics/random-forest