Maskininlärning

Teoretiska frågor, och ett exempel

Dan Heikenberg

EC Utbildning

Examensarbete- Maskininlärning

202403

Abstract

Denna rapport består av flera delar. Teoretiska frågor om maskin-inlärning kommer att besvaras i teori-sektionen. Det kommer även att inkluderas en utvärdering av ett maskin-inlärningsexempel. Maskin-inlärningen kommer att utföras med ett data set kallat MNIST, vars data representerar siffror. Ett försök att skapa en streamlit-app med den bästa modellen, kommer även att göras. Rapporten kommer slutligen att inkludera en självutvärdering.

Skapas automatiskt i Word genom att gå till Referenser > Innehållsförteckning.

Innehållsförteckning

1 Inledning

Maskin-inlärning är en viktig del i jobbet som Data Scientist, och så även rapport-skrivning. Att kunna ta del av ett data-set, bestämma vilken typ av modell som ska testas, och kunna testa och justera denna för ett så bra resultat som möjligt, är något som bör övas på. Detta resultatet är dock värdelöst om det inte kan förmedlas, vilket belyser rapport-skrivningens viktiga del i sammanhanget. Framtida arbetsgivare måste kunna se tillvägagångssätten samt resultaten, så att de kan få en överblick, men också kunna gå djupare om de vill. Det kan också vara bra för data scientisten att ha en rapport som ett tydligt dokument över något som kan komma att ifrågasättas eller rådfrågas längre fram när minnet inte är lika färskt.

Syftet med denna rapport är att öva både på rapport-skrivning, och att öva på maskin-inlärning. För att uppfylla syftet så kommer följande frågeställning(ar) att besvaras:

1. Hur kan jag använda maskin-inlärning till ett givet dataset - MNIST?
2. Hur kan jag skapa en streamlit-app som kan göras en modell mer tillgänglig?
3. Hur tycker jag att arbetet, inklusive rapport-skrivningen har gått?

2 Teori

2.1 Klassifikationsmodeller

Det tilldelade datasetet är ”MNIST”, vilket betyder att jag kommer att använda mig av klassificerings-modeller, då det är ”supervised learning” med ”labels”, men till skillnad från regressions-problem så behöver klassificering ske till olika klasser (1, 2, 3 o.s.v.).

2.1.1 SVC

Support Vector Classifier är en modell jag nyligen lärt mig finns, och det ska bli intressant att testa den. Den delar upp data i olika klasser med hjälp av vektorer som den drar mellan data-punkterna på bästa möjliga sätt.

2.1.1.1 Hyperparametrar

Det finns hyperparametrar som kan justeras för att få ett bättre resultat, bland annat vilken typ av ”kernel”, om det ska vara linjära vektorer eller inte.

2.1.2 Random Forest Classification

Detta är en modell som jag sett användas i många exempel, och som skulle kunna ge ett bra resultat. Den använder sig av ”besluts-träd”. Vid varje ”träd/gren” sker en bedömning, och till sist görs en slutgiltig bedömning av alla resultat.

2.1.2.1 Hyperparametrar

Det finns hyperparametrar som kan justeras för att få ett bättre resultat, bland annat n_estimators som bestämmer ”djupet” i ett ”träd”. Detta kan vara bra att justera då för lågt eller för högt värde kan leda till under/overfitting.

2.1.3 Accuracy Score

Accuracy score returnerar 1, enligt Scikit-Learn (Sci-kit Learn, n.d.) om resultatet är perfekt. 0.95 skulle betyda 95%.

2.2 Streamlit

Med streamlit kan man med python skapa en app som blir mer tillgänglig för exempelvis kunder.

3 Metod

Datan har erhållits via fetch_openml, vilket är dataset som är en del av sklearn. Med programmet Jupyter har jag sedan delat upp datan, både i ”features”, samt till test, validate och tränings-set. Därefter har jag tränat modellerna med tränings-setet, validerat dem med validate-setet, och slutligen har jag valt den med högst accuracy score, och testat den mot test-setet. Modellerna har även fått sina hyperparametrar justerade för att se om det var möjligt att förbättra resultatet.

En app ska skapas i streamlit, där den bästa modellen ska kopplas till bilder som användaren ska kunna ladda upp/fånga via webbkamera. Med python-kod omvandlas bilderna till rätt storlek (28x28 pixlar), vilket blir rätt antal data-punkter som modellerna tränats på. Med en funktion så delas ljus-styrkans spektrum i bilden till antingen 0 eller 1, för att skapa en svart-vit version. Datan läggs sedan ut i en rad så att den kan matas in i modellen som laddats in i strreamlit.

4 Resultat och Diskussion

Accuracy score för olika modeller	
SVC Linear C=10 (default)	0.912
SVC Poly gamma=1 C=0.1	0.9605
Random Forest Classifier (default)	0.9495
Random Forest Classifier max_features=”sqrt, n_estimators=150	0.9505
SVC Poly gamma=1 C=0.1 (predict på test-data)	0.96

Tabell 1: Accuracy score för de fyra valda modellerna.

SCV fick bättre resultat av att inte ha sina ”default” parametrar, och även om Random Forest Classifier fick ett högt värde utan att justeras, så blev det inte mycket bättre resultat efter justering.

Med den justerade SVC-modellen och test-datan, kan man även i en confusion matrix se att den gissar ganska bra, förutom relativt på (2, 7) och (2, 8).

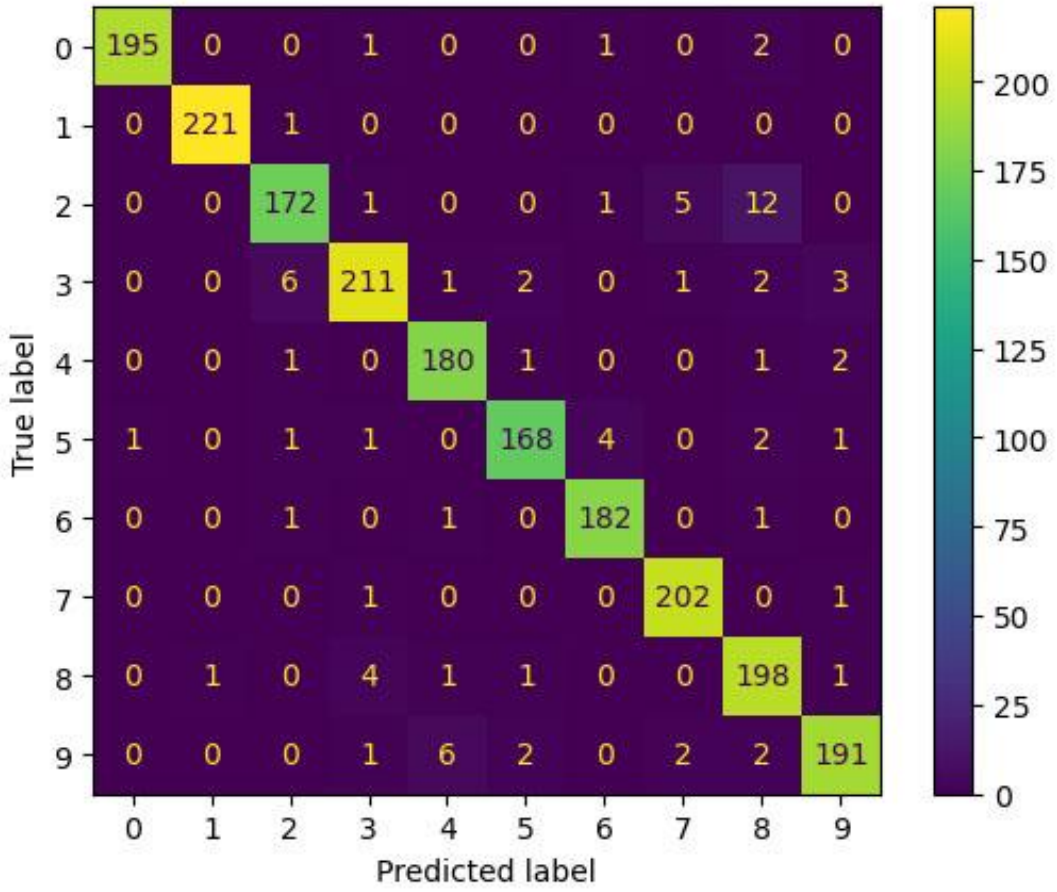


Bild 2: Confusion matrix över SVC justerad med prediction mot test-datan.

Streamlit-modellen fungerar inte, och jag vet inte varför. Vad jag än testar, så predictar den ”8”.

När jag testar med en MNIST-bild, föreställande ”7”, så kan jag tydligt i min kod se ”7” representerat, men likväl tolkar modellen det som ”8”, och jag kan inte lista ut vad problemet är, vilket är synd för det känns som jag kom ganska långt.

Bild 2 och 3: Bild 3 visar ”7” som med kod konverterats till ettor och nollor, och man kan i bild 2 se ”7” representerad.

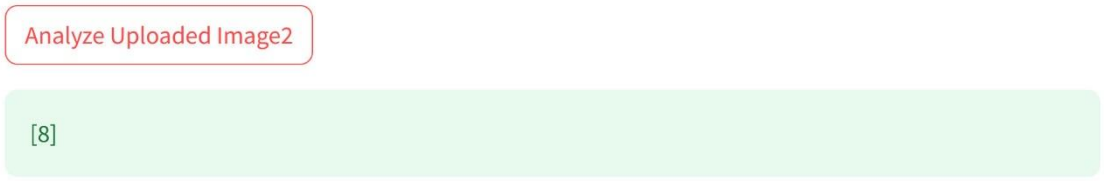


Bild 4: Screenshot från streamlit appen där ”7” från bild 3 konverterats till vad som syns på bild 2, och sedan tolkas av modellen som ”8”.

5 Slutsatser

Genom att testa olika maskininlärnings-modeller i programmet Jupyter, samt justera deras hyper-parametrar, kunde jag nå fram till en modell som skulle kunna användas till datan med relativt hög säkerhet.

Det hade kunnat gå att lägga mer tid på justering, och det hade kunnat ge ett ännu högre resultat, men till just denna uppgiften var jag nöjd, och även lite förvånad över hur mycket bättre SVC-modellen blev efter justeringen.

Rapport-skrivningen har gått bra, och har fungerat som en introduktion. Med mer tid och mer material från ett större arbete, skulle rapporten fyllas ut mer.

Det har varit lite problematiskt att göra rapporten samtidigt som jag arbetar på Streamlit-appen. Fungerar inte appen så har det inte känts lönt att skriva med den i rapporten, men i sista stund ändrade jag mig trots att jag inte fick appen att fungera, eftersom det även är en kunskapskontroll.

6 Teoretiska frågor

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

Tränings-data är datan som används för att träna ”fit” modellen. Den förbereds först av data scientisten så att en model ur passande kategori kan läsa och lära sig av den.

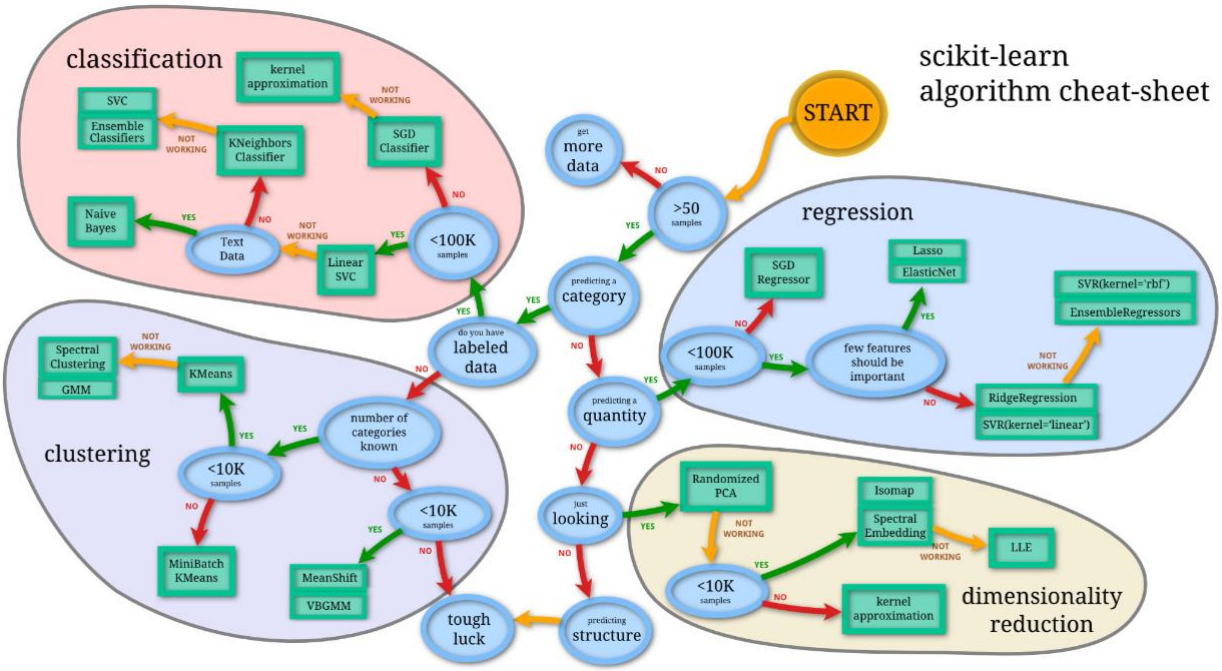
Med ett evaluation-dataset kan man få en unbiased evaluation av modellen som använts på train-setet, medan man också kan ställa in hyperparametrarna (towardsdatascience, n.d.) för ett ännu bättre resultat. Detta gör att man har ett helt orört ”unbiased” test-set till senare, som inte hjälpt till att ställa in hyperparametrarna.

Tillsist använder man test-setet för att få en unbiased evaluation av den slutgiltiga tränad hyperparameter-inställda modellen. Det är slutresultatet, men man kan fortfarande göra om allt med andra modeller.

6.2 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”?

Julia kan använda sig av cross-validation för att se hur modellerna presterar (Censius, n.d.). Hon kan sedan räkna ut medelvärdet av det antal iterationer som väljs, och jämföra det med medelvärdet av de andra modellerna.

Hon kan även se över sin data och fundera över vilken modell som är lämpligast att använda. Ska få eller många features vara viktigt? Linjär regression är den mest basic versionen av linjär regression, men som fortfarande kan ge bra resultat. Modellen kan lägga mycket vikt på en enda feature, vilket dock kan leda till overfitting i små dataset (towardsdatascience, n.d.). När alpha-hyperparametern i Lasso-modellen, närmar sig 0, så börjar den likna linjär regressions-modellen (medium, n.d). Lasso-modellen är alltså en utveckling av linjär regression.



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

Ett regressionsproblem är något som kan utredas genom att analysera relationen mellan en dependent variable (target) och independent variables (predictors) (geeksforgeeks, n.d.). Det finns ett continuous värde som ska predictas, t.ex hur stor lönen är, beroende på t.ex. utbildning och anställningstid, eller hur stor skörden blir, beroende på mängden regn, eller åldern på en person, o.s.v.

Regressions modeller kan evalueras med hjälp av root mean squared error, vilket classification modeller inte kan (machinelearningmastery, n.d.).

Random Forest Regression är en modell som jag personligen tycker mig ha sett vara populär. Den skapar flera ”decision trees” som tränas med olika subsets” av träningsdatan. Ett medelvärde av alla resultat, blir det slutgiltiga resultatet (geeksforgeeks, n.d). Får man t.ex. problem med overfitting, så kan man testa t.ex. lasso regresson.

När bara ett litet antal predictors är viktiga, så tenderer Lasso att prestera bättre än Ridge, och när det istället är ett större antal, så kan Ridge fungera bättre (statology, n.d.)

6.4 Hur kan du tolka RMSE och vad används det till: $RMSE = \sqrt{\sum(yi - \hat{y}i)}\ 2\ i$

I regressions-problem kan man använda sig av root mean squared error för att utvärdera hur stort fel (eller rätt) det blir när modellen predictar datan. Det som räknas fram är avståndet mellan det gissade värdet, och det egentliga värdet. Root och squared är för att negativa värden ska förvandlas till positiva värden, så att de inte annulerar de positiva värdena och ger en felaktig bild av att ett

positivt värde plus ett engativt värde innebär ett perfekt värde. Mean innebär medelvärdet av alla dessa värdena, så att det blir ett samlat resultat.

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

Klassifikeringsproblem är fortfarande inom området ”supervised learning”, med ”labels”, men skillnaden från regressionsproblem är att klassifikering klassificerar till olika klasser, t.ex. hund eller katt. Tillämpningsområden kan t.ex. vara inom sjukvården, för att undersöka om patienter har en sjukdom eller inte.

Olika modeller är Random Forest Classifier, K-Nearest Neighbors, och Logistic Regression (trots det lite missvisande namnet).

Med en confusion matrix kan man tydligt se hur modellen presterat, genom att se sanna positiva, falska positiva, sanna negativa och falska negativa prediktioner i binära (1 och 0) klassifikationer. Med confusion matrix kan man få fram t.ex. accuracy, precision, recall och f1, vilket är olika sätt att se hur modellen presterar, och kan vara viktigt beroende på t.ex. sjukdoms-prediktion då man vill veta hur mycket falska negativa o.s.v. som kan tillåtas. Confusion matrix kan även användas till mer än binär klassificering (Geeks For Geeks, n.d.).

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

K-Means används inom unsupervised machine learning, och ä ren algoritm som skapar cluster av datan (Stackabuse, n.d.). ”K” står för antal kluster, och från de punkter som kommer att representera ett kluster, så mäts avståndet från mellan dessa och alla data-punkter för att se vilka datapunkter som tillhör vilket kluster.

K-means är ett sätt att gruppera data-points ()StatisticsByJim, n.d.) K-means kan användas för att t.ex. se om det finns olika grupperingar av människor med olika beteenden beroende på andra faktorer såsom t.ex. lön. Det kan t.ex. finnas ett kluster av människor inom ett visst löne-intervall som verkar föredra en viss bil-modell.

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

Med ordinal encoding så tilldelas varje unik kategori med ett stigande INT-värde (1, 2, 3 o.s.v). Detta kan dock vara missvisande om det är kategorier som inte är naturligt stigande, alltså nominala variabler, för det kan peka på ett förhållande som inte finns (Machine Learning Mastery, n.d).

One-hot encoding är också ett sätt att omvandla kategorisk data till numerisk, så att det faktiskt kan användas av en modell. Med one-hot encoding så tilldelas alla kategorier ”1” eller ”0” i vars en variabel som representerar t.ex. den färgen, och man kan då undvika problemet ovan.

One-hot encoding är inte helt optimerad, då den sista färgen inte måste representeras på samma sätt som den andra variablerna för att modellen ska kunna lista ut att det är en annan färg. Om den sista färgen helt enkelt representeras av att ingen av de andra färgerna har ”1” där, så räcker det att alla andra färgers ”0”-värde representerar den sista färgen.

Det finns även linear regression-modeller som kräver just dummy variable encoding för att fungera (Machine Learning Mastery, n.d.).

6.8 Göran påstår att datan antingen är ”ordinal” eller ”nominal”. Julia säger att detta måste tolkas. Hon ger ett exempel med att färger såsom {grön, röd, grön} 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?

Med en snabb tolkning skulle jag säga att färger är nominala, röd, grön, eller katt, hund. Det är inte ordinal data som t.ex. dagis, lågstadieskola, högstadieskola, gymnasium. Samtidigt är inte allt alltid svart/vitt. Allt kan sättas i olika sammanhang, och människor kan sätta olika värde på kategorier. Någon kanske tycker att hundar är bättre än katter, och det finns favoritfärger. Grönt kan nog tänkas vara mer positivt än rött i tävlingssammanhang, men på en fest där temat är ”röd klädsel” så kan det vara dåligt att klä sig i gröna kläder. Jag tror att man får titta på kontexten för att fatta det bästa beslutet.

6.9 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?

Streamlit är ett framework som kan hjälpa att skapa och dela en interaktiv python app där man kan presentera data, och även t.ex. träna och göra modeller tillgängliga för praktisk användning.

7 Självutvärdering

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

   Streamlit var en rolig utmaning, även om den var svår. Skillnaden att ladda en bild, att ”ladda upp” en bild, och att ta en bild med webbkameran, var större än jag trodde. Jag har listat ut varje steg jag behöver ta, och löst varje steg för sig. När jag t.ex. behövt ta reda på hur man minskar storleken på en bild utan att förstöra bildens information för mycket, så har det gått att söka sig till.

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

   Jag kan bara få min Streamlit, med min modell, att predicta ”8”. Fungerar den inte så är det inte VG enligt uppgiften, men jag ser väldigt gärna ett fungerande exempel efter kursens slut. Jag kan se att min kod lyckas omvandla bilder till ettor och nollor som representerar bilderna, vilket jag visar här i rapporten, men av någon anledning vill inte modellen predicta rätt. På något vis måste jag nog lista ut varför.

3. Något du vill lyfta fram till Antonio?

   Streamlit-uppgiften var rolig även om jag inte klarade den! Det är skoj när man knyter ihop saker, Streamlit-app, med modellen man tränat, med python-kod som man måste skriva för att länka ihop allt.

Källförteckning

Censius. Machine learning model selection techniques. Hämtad 17 mars, 2024, från Celsius sida https://censius.ai/blogs/machine-learning-model-selection-techniques#blogpost-toc-2

Geeks for geeks. Confusion matrix. Hämtad 17 mars, 2024, från Geeks for geeks sida https://www.geeksforgeeks.org/confusion-matrix-machine-learning/

Geeks for geeks. Regression in machine learning. Hämtad 17 mars, 2024, från Geeks for geeks sida https://www.geeksforgeeks.org/regression-in-machine-learning/

Machine learning mastery. Classification versus regression in machine learning. Hämtad 17 mars, 2024, från Machine learning mastery´s sida https://machinelearningmastery.com/classification-versus-regression-in-machine-learning/

Machine learning mastery. One hot encoding for categorical data. Hämtad 17 mars, 2024, från Machine learning mastery´s sida https://machinelearningmastery.com/one-hot-encoding-for-categorical-data/

Scikit learn. Sklearn metrics accuracy score. Hämtad 17 mars, 2024, från Scikit learns sida sklearn.metrics.accuracy_score — scikit-learn 1.4.1 documentation

Statistics by jim. K means clustering. Hämtad 17 mars, 2024, från Statistics by jim´s sida https://statisticsbyjim.com/basics/k-means-clustering/

Statology. When to use ridge lasso regression. Hämtad 17 mars, 2024, från Statology´s sida https://www.statology.org/when-to-use-ridge-lasso-regression/

Stackabuse. K means clustering with scikit learn. Hämtad 17 mars, 2024, från Stackabuse's sida https://stackabuse.com/k-means-clustering-with-scikit-learn/

Tahera Firdose Medium. Hygienrutiner och klädpolicy. Hämtad 17 mars, 2024, från Tahera Firdose Medium ´s sida https://tahera-firdose.medium.com/lasso-regression-a-comprehensive-guide-to-feature-selection-and-regularization-2c6a20b61e23

Towards data science. Whats the difference between linear regression lasse ridge and elasticnet. Hämtad 17 mars, 2024, från Towards data science´s sida https://towardsdatascience.com/whats-the-difference-between-linear-regression-lasso-ridge-and-elasticnet-8f997c60cf29

Towards data science. Train validation and test. Hämtad 17 mars, 2024, från Towards data science´s sida https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7