Machine learning

En komparativ rapport om Random Forest Classifier och Support Vector Machine



Nil Abukar

EC Utbildning

Machine Learning

2024-03-21

# Abstract

This report focuses on a comparative study between Random Forest Classifier and Support Vector Machine based on MNIST-dataset. The report aims to find the differences in test accuracy with both classifiers. The result shows that Random Forest Classifier has better test accuracy and predictions compared to Support Vector Machine. However, the diffrences are not as big as it would have been due to their test accuracies are relatively close.

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

Innehållsförteckning

[Abstract 2](#_Toc162025051)

[1 Inledning 1](#_Toc162025052)

[2 Teori 2](#_Toc162025053)

[2.1 Random Forest Classifier 2](#_Toc162025054)

[2.2 Support Vector Machine 2](#_Toc162025055)

[2.3 Beslutsträd 3](#_Toc162025056)

[2.4 ROC and AUC kurva 3](#_Toc162025057)

[3 Metod 5](#_Toc162025058)

[4 Resultat och Diskussion 6](#_Toc162025059)

[5 Slutsatser 9](#_Toc162025060)

[6 Teoretiska frågor 10](#_Toc162025061)

[7 Självutvärdering 12](#_Toc162025062)

[Källförteckning 14](#_Toc162025063)

# Inledning

Ge en bred bakgrund till ditt arbete och varför det är relevant. Smalna successivt av och koppla bakgrunden till detta arbete. Fortsätt med syfte och frågeställning (syfte och frågeställning kan du ha i en egen underrubrik om du önskar det).

Syftet med denna rapport är komparativ studie mellan Random Forest Classifier och Support Vector Machine, för att uppfylla syftet så kommer följande frågeställning(ar) att besvaras:

1. Vilken classifier visar närmast korrekta resultat?
2. Hur skiljer sig dessa classifier sig åt i testning och predikteringen?

# Teori

I denna del av rapporten kommer att handla om teoretiska koncept gällande valda metoder för min datamodellering. Rapporten är en komparativ studie, det vill säga en jämförelse sker mellan olika klassifikationer som i detta fall Random Forest Classifier och Support Vector Machine. I datamodelleringen har det dock använts beslutsträd och ROC och AUC kurva men det har dock inte gjorts en jämförelse av alla fyra modeller utan de två först nämnda.

## Random Forest Classifier

Random Forest är en populär maskininlärnings algoritm. Det funkar bäst att hantera stora och komplexa datasets så väl som hantera hög dimensionella funktioner och tillhandahålla insikt i funktionens vikt. Denna algoritms förmåga att upprätthålla en hög prediktiv riktighet medan det minimerar överanpassning vilket gör det till ett populär val inom olika sektorer så som finans, vården och bildanalyser för att nämna få (GfG. (2024, January 31).

Det kan tillämpas för att lösa regression (numeriska variabler) och klassifikation (kategoriska variabler) problem. Det är en metod i sin helhet som kombinerar prediktioner från andra modeller. För varje liten modell i en Random Forest är ett beslutsträd.

I en Random Forest klassifikation är många beslutsträd skapade genom att använda olika slumpmässiga delmängder av data och funktioner. Varje beslutsträd ger sin åsikt om hur att klassifiera data. Å andra sidan, prediktioner är genom kalkylationer för varje beslutsträd och tar det mest populära resultatet medan inom regression används prediktioner en genomsnittlig teknik (Shafi, A. (2023, February 24*).*

Random Forest Classifier kan hantera både klassifikation och regression och har en förmåga att tillhandahålla funktionens viktiga resultat för funktioner som gör det ett värdefullt verktyg för att förstå betydelsen av olika variabler i datamängden.

Detta verktyg är anpassad för att framhäva noggrannheten och dess robust i klassifikationsuppdrag. Det bygger en hel del av beslutsträd under träning och varje beslutsträd är byggd på att använda delmängder av tränade data och slumpmässig delmängd av funktioner som introducerar mångfald bland träden. Detta leder till att modellen blir mer robust och mindre benägen till att överanpassa (GfG. (2024b, January 31*).*

## Support Vector Machine

Support Vector Machine (SVM) är en kraftig och mångsidig algoritm som i kärna kan avgränsa optimala hyperplan i ett högdimensionellt utrymme, och effektivt separera de olika klasser i en datauppsättning (Andreoni, R. (2023, October 12). Den är kapabel till att utföra linjär och icke linjär klassifikation (hands on machine learning) . Dess effektivitet är inte begränsad till klassificeringsuppgifter utan det är även väl lämpad för regression (Andreoni, R. (2023b, October 12).

SVM söker efter det bästa beslutet som separerar två klasser med den högsta generaliseringsförmågan. SVM vill det minsta avståndet mellan data punkter och beslut att vara det största som möjligt vilket skiljer logistisk regression.

SVM kan inte tillhandahålla en estimerad sannolikhet då dessa är kalkylerade att använda en five-fold cross validation (Zhang, Z. (2021, December 11). När man skapar en SVM modell genom Scikit-learn och man kan specifiera ett antal av hyperparametrar, t.ex. C hyperparametern (Géron, A. (2019*).*

En linjär SVM classifier är effektiv och kan arbeta väldigt väl i många fall även om många datamängder är inte nära till att bli linjärt separerat. För att kunna hantera icke linjära datamängder är att tillsätta flera funktioner så som polynomial funktioner och det kan resultera i en linjär separat datamängd (Géron, A. (2019*).*

## Beslutsträd

Beslutsträd är inlärningsalgoritm där data kontinuerligt delas upp i mindre delar till den når sin klass. Detta verktyg är en av de enklaste och populära klassifikations algoritmer för att kunna förstå och tolka data (Chauhan, N. S. (n.d.). Den använder terminologier så som ’nodes’, ’edges’ och ’leaf nodes’. Beslutsträdet talar om för oss efter beräkning av databasen som i sin tur talar om för oss hur stor osäkerhet som finns i databasen. Desto mindre osäkerheten är desto bättre klassificerings resultat får man (Yassin, N. I., Omran, S., Houby, E. M. F. E., & Allam, H. (2018).

Syftet med beslutsträd är att skapa en träningsmodell som kan användas för att prediktera klassen eller värdet av en variabel genom att lära sig enkla beslutsregler från tidigare data, det vill säga träningsdata (Chauhan, N. S. (n.d.)

Beslutsträd använder ett flödesschema som är strukturerad som ett träd för att visa predikteringarna resultatet från en serie av olika funktion baserade splittringar. Det startar från root node och slutar med ett beslut taget av löven. En beskrivning av terminologier kommer att härefter göras detta för att det ska underlätta förståelsen av beslutesträdets funktioner. Root node är början av ett beslutsträd där hela populationen eller datasetet startar att dela upp sig baserat på olika funktioner eller tillstånd. Decision nodes däremot är nodes som resulterar från splitting root nodes. Dessa nodes representerar mellan beslut och tillstånd inom trädet. När man splittrat nodes ytterligare tills att det inte är möjligt oftast indikerar på slutlig klassifikation eller resultat, som även kallas för Leaf Nodes så väl som slutliga nodes (Saini, 2024)

## ROC and AUC kurva

ROC är en sannolikhetskruva och AUC representerar graden eller måttet på särskiljandet. Den talar om hur bra modellen kan skilja sig från klasserna och desto högre AUC desto bättre är modellen att prediktera 0 klasser som 0 klasser och 1 klasser som 1. ROC är däremot plottad med True Predictions på y-axeln mot False Predictions på x-axeln.

Hur ska man tolka datas utförande? En utmärkt modell har AUC nära till 1 vilket innebär att den har goda mått av särskiljhet. En dålig modell om har AUC nära till 0 betyder att den har värsta måtten av särskiljhet och om AUC är 0.5 innebär det att modellen har inget utrymme att klass separera (Narkhede, 2022)

# Metod

Min maskininlärningsarbete har utförts med hjälp av MNIST-datasetet som består av handskrivna siffror. Vid en datainsamling och förberedelser har MNIST-datasetet laddas in som i sin tur delar upp data i tränings- och testuppsättningarna så väl som pixelvärdena normaliseras för att det ska kunna ligga mellan 0 till 1.

Jag har valt använda tre klassificeringsalgoritmer, Random Forest (RF), Support Vector Machine (SVM) och Decision Tree (DT) i min modellträning. Modellerna tränas med träningsdata och utvärderas senare med hjälp av test data. I rapporten kommer jag att använda mig av accuracy (testnoggrannhet) som utvärderingsmått. Vidare kommer det under analysen att presenteras antalet korrekta och felaktiga predikteringar för varje klass och det kommer även att presenteras skillnader i predikteringar mellan RF och SVM. Detta kommer att visualiseras med hjälp av slumpmässigt valda exempel och en jämförelse av predikteringarna. Jag hade kunnat välja specifika siffror men i detta fall var det intressantare att se slumpmässigt valda siffor och vad predikteringarna har varit samt dess sanna ’’värde’’

En ROC och AUC kurva beräknas för att bedöma RF och SVM prestation lite mer ytterligare. Genom denna metod och visualiseringar kan man utvärdera och jämföra prestationerna hos olika klassificeringsalgoritmer för att kunna hitta det som passar bäst. I mitt fall är detta en komparativ studie där olika klassificeringsalgoritmer jämförs och studeras.

# Resultat och Diskussion

Min kod utför en omfattande analys av två maskininlärningsalgoritmer, RF och SVM på en MNIST-dataset som består av handskrivna siffror. Koden visar en komplett maskininlärningsgrund för att träna och utvärdera två olika klassificeringsalgoritmer. Datan har normaliserats och delats upp i två uppsättningar av träning och test och i sin tur krossvalidering för att beräkna noggrannheten. Det är för att det ska prediktera för både RF och SVM och utvärdera dem med olika prestationsmått som noggrannhet, precision, recall och F1. Därtill har det skapats visualiseringar för att klassificeringarnas prestation och identifiera skillnaderna i respektive prediktion.

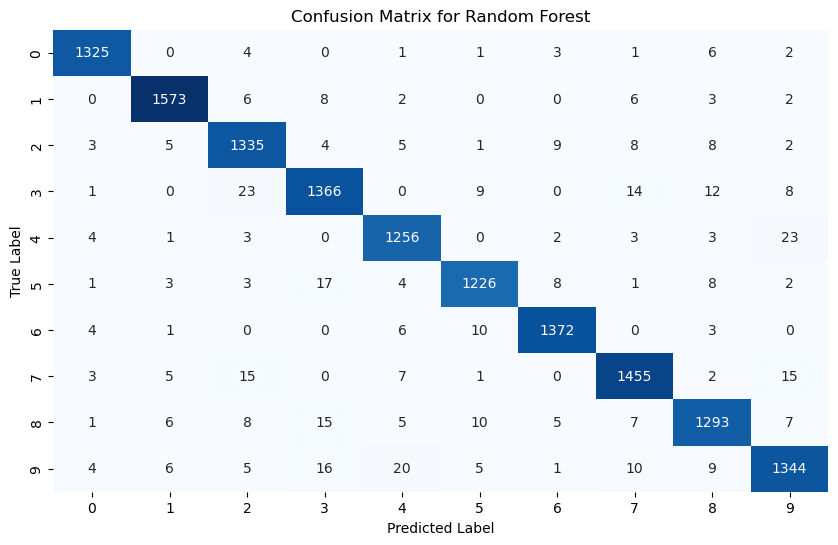
Test noggrannheten för RF visar cirka 97 procent medan SVM gav cirka 94 procent och Decision Tree cirka 87 procent. Det visar hur RF och SVM presterar på oberoende testdata och det är andelen korrekta predikteringar som modellen gör på testdatan. Det innebär att RF predikterar 97 procent av de exempel som ingår i test datan, det vill säga 97 av 100 prediktioner är korrekta. Detta innebär ungefär samma för SVM där det är 94 av 100 och Decision Tree 87 av 100 prediktioner.

En bild som visar text, linje, Graf, skärmbild

Automatiskt genererad beskrivning

Test noggrannheten visar att RF presterade bäst med noggrannheten vilket ovan demonstreras i bilden ovan där RF växer stegande medan SVM har varit rak ungefär vid 0,2 börjar den att sjunka. Detta leder till att bekräfta att RF är högst troligt den bäst lämpade i detta fall med test noggrannheten.

Oaktad träning och test av tre klassificeringar har jag valt att studera RF och SVM eftersom deras höga antal prediktioner. Det har identifierats skillnader mellan de två klassificeringarna genom att jämföra deras confusion matrix och det har visats på vilka instanser de skiljer sig åt. I RF är 1 325 antalet korrekta prediktioner de antalet korrekta prediktionerna och att den gör 1 573 felaktiga prediktioner. Färgerna i matrisen visar antalet korrekta och felaktiga. Mörka färger efter den första vid 0 i vänstra hörn indikerar vanligtvis att modellen gör många felaktiga prediktioner, vilket i detta fall blir 1 572.



För SVM visar confusion matrixen att antalet korrekta predikteringar är 1 310 och antalet felaktiga är 1 571. I båda matriser är det tydligt att vid sanna värdet är 1 det oftast blir felaktiga predikteringar men även att vid 7 kan en hög andel av felaktiga predikteringar.

En bild som visar text, skärmbild, Rektangel, diagram

Automatiskt genererad beskrivning

Figur 1: Hur man lägger in tabell eller figur nummer samt beskrivning.

Det är nog den största utmaningen hade kunnat vara att använda sig av hyperparametrar för båda klassificeringar och hantering av eventuell överanpassning. Jag försökte tillämpa GridSearchCV i min kod dock stötte jag på svårigheter med att ladda in min data och därmed fick det lämnas utanför med risk att prestationen inte blir bra.

Men trots valen som har gjorts i datamodelleringen visar goda resultat utifrån hur klassificeringarna har presterat utan hyperparameterar. Enligt predikteringarna utifrån randomiserade siffror visar att RF prediktioner matchar med det sanna värdet.

En bild som visar text, Teckensnitt, nummer, Grafik

Automatiskt genererad beskrivning

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Index | True label | RF prediktion | SVM prediktion |
| 0 | 2 | 8 | 8 | 5 |
| 1 | 9 | 4 | 4 | 7 |
| 2 | 11 | 9 | 9 | 7 |
| 3 | 46 | 2 | 2 | 8 |
| 4 | 49 | 9 | 4 | 9 |
| 5 | 78 | 5 | 5 | 6 |
| 6 | 84 | 7 | 7 | 2 |
| 7 | 86 | 1 | 1 | 9 |
| 8 | 100 | 3 | 3 | 5 |
| 9 | 102 | 0 | 0 | 5 |

Det kan vidare argumenteras för att RF är den bäst lämpade klassificering algorimten i detta fall då träning och testning så väl som predikteringar har visat en otrolig noggrannhet till det sanna värdet. Detta leder till att ROC curvan som bekräftar ovannämnda uppgifter om RFs prestation att AUC curvan är lika med 1.00 på alla instanser medans för SVM pendlar den mellan 0.97-1.00. Detta talar för att RF modellen är utmärkt i förhållande till SVM.

# Slutsatser

Mina frågeställningar var vilken classifier visar närmast korrekta resultat och hur de skiljer sig i testning och prediktering.

Det har visat att vara RF som är närmast korrekta resultat med en test noggrannhet på cirka 97 procent och en AUC på 1.00 på e ROC curva talar för att det är en utmärk modell. Det är viktigt att ha i åtanke att det inte skiljde sig så mycket mellan RF och SVM då det skiljde sig tre procentenheter i test noggrannheten. Det har dock resulterat i att predikteringen för SVM var lite sämre än RF vilket kan eventuellt beror på procentenheten i test noggrannheten.

Sammanfattningsvis, finner jag att jag har fått svar på mina frågeställningar genom min kod. Jag hade kunnat vara mer påhittig och nyfiken för att få en djupare förståelse för modellernas prestation. Även kunna identifiera möjligheter till förbättring och justeringar för att öka dess effektivitet och tillförlitlighet. Det hade varit intressant att kunna studera detta mer på djupet och använda sig av hyperparametrar samt olika klassificeringsalgoritmer.

# Teoretiska frågor

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

Svar: Träningsdelen av ett datasett används för att träna modellen. Syftet är att modellen ska lära sig mönster och egenskaper i data som gör att den kan göra korrekta prediktioner på nya och obekant data. Valderingsdelen används för att finjustera modellens hyperparamentrar och bedöma dess prestationer under träningsprocessen. Testdelen används för att utvärdera modellens prestationer efter att den har tränats (träning) och finjusterats (validering). Genom att dela upp dataseten så som Kalle har gjort kan man utveckla och utvärdera modeller på ett sätt som minimerar risken för overfitting (överanpassning) samt ger en tillförlitlig bedömning av dess prestationer.

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 ska hon välja vilken av de tre modellerna skall hon fortsätta använda när hon inte skapat ett explicit ’’validerings-dataset’?

Svar: Hon kan använda sig av cross-validation för att göra en jämförelse och välja den bästa modellen.

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

Svar: Det är en form av maskininlärningsproblem där syftet är att prediktera kontinuerliga eller numeriska värden utifrån en uppsättning av imatningsvariabler. Det finns linjär, lasso och random forest regression som kan tillämnas inom ekonomi, medicin, marknadsföring för att nämna få.

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

Svar: Root Mean Squared Error (RMSE) är två huvudsakliga indikationerna för en regressionsmodell. Den mäter det genomsnittliga värdet som predikterats av en model och dess faktiska värden. Det ger en uppskattning på hur bra modellen kan prediktera, det vill säga noggrannheten. Ju lägre RMSE är desto bättre är modellen.

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

Svar: Det innebär att syftet är att prediktera en diskret eller kategorisk utgångsvariabel utifrån en uppsättning av inmatningsvariabler. Några exempel är Logistisk regression, beslutsträd etc som kan tillämpas inom handel, medicin för att nämna få.

En confusion matrix är en tabell som används för att utvärdera prestationerna hos en klassificieringsmodell. Det visar antal korrekta så väl som felaktiga prediktioner och ger en mer delaterad bild av modellens förmåga att korrekt klassificera observationer i olika klasser. Detta delas in i fyra grupper TP (true positive), TN (true negative), FP (false positive) och FN (false negative).

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

Svar: K-means modellen är en klustring metod som delar upp en datamängd i k-nära eller kluster. Modellens syfte är att minimera avståndet mellan datapunkter inom samma kluster samtidigt som det ökar avståndet mellan kluster. Detta kan användas exempelvis inom marknad för kundanalys.

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

Svar: Ordinal encoding är en teknik som omvandlar kategoriska funktioner till ett numerisk format. One hot encoding är en process som kategoriska variabler som konverteras till en ny kategorisk kolumn och tillger ett binär värde. Dummy variable, tillika indicator variabel, är en som tar ett binärt värde för att indikera närvaron eller frånvaron av några kategoriska effekter som kan förväntas ändra resultatet.

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?

Svar: Båda har rätt i sina respektive argument. Göran hävdar att datan är antingen ordinal eller nominal vilket är korekkt då det är två huvudtyper av kategoriska variabler. Julia har även korrekt i sin sak eftersom färgerna inte kan rangordnas på ett meningsfullt sätt. Julias exempel på en röd skjorta kan symbolisera skönhet, då kan röd anses vara ordinal eftersom det finns en inneboende ordning eller rangordning i detta sammanhang.

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

Svar: Streamlit är en open-source som används för att skapa användarvänliga applicationer i Python.

# Självutvärdering

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

* Det har varit en stor databas och att det kunde göra laddning av datan långsam. En annan utmaning var att hitta frågeställningar som var intressant för mig

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

* Jag anser att jag bör få G då jag har nått samtliga kriterier för att nå G i betyg.

1. Något du vill lyfta fram till Antonio?

* Tack för denna kurs. Det har varit intressant och tycker videoinspelningarna har varit lärorika.

# Källförteckning

Andreoni, R. (2023, October 12). *Support Vector Machine with Scikit-Learn: A Friendly Introduction*. Medium. <https://towardsdatascience.com/support-vector-machine-with-scikit-learn-a-friendly-introduction-a2969f2ff00d>)

Andreoni, R. (2023b, October 12). *Support Vector Machine with Scikit-Learn: A Friendly Introduction*. Medium. <https://towardsdatascience.com/support-vector-machine-with-scikit-learn-a-friendly-introduction-a2969f2ff00d>)

Chauhan, N. S. (n.d.). *Decision Tree algorithm, explained - KDnuggets.* KDnuggets. https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html

Géron, A. (2019). *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O’Reilly).

GfG. (2024, January 31). *Random Forest Classifier using Scikit-learn. GeeksforGeeks.* <https://www.geeksforgeeks.org/random-forest-classifier-using-scikit-learn/>.

Saini, A. (2024, January 5). *Decision Tree – a Step-by-Step guide.* Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2021/08/decision-tree-algorithm/>

Shafi, A. (2023, February 24). *Random Forest Classification with Scikit-Learn*. <https://www.datacamp.com/tutorial/random-forests-classifier-python>

Yassin, N. I., Omran, S., Houby, E. M. F. E., & Allam, H. (2018*). Machine learning techniques for breast cancer computer aided diagnosis using different image modalities: A systematic review*. Computer Methods and Programs in Biomedicine, 156, 25–45. <https://doi.org/10.1016/j.cmpb.2017.12.012>

Zhang, Z. (2021, December 11). *Support vector machine explained - towards data science. Medium*. <https://towardsdatascience.com/support-vector-machine-explained-8bfef2f17e71>