Maskininlärning

Kunskapskontroll 2



Mikaela Bennshagen

EC Utbildning

2024 – 03

# 

# Abstract

This report describes and analyzes which Machine Learning model or models that gives the best prediction when using the MNIST database. The four models that were evaluated were Logistic Regression, K-Nearest Neighbors, Support Vector Machine and Random Forest. Different hyperparameters were also tested along the models, but were ultimately not used as they did not enhance the performances of the models. Even though all models performed well (all over 90%) it was the Random Forest model that predicted most reliably.

Innehållsförteckning

[Abstract 2](#_Toc162005538)

[1 Inledning 1](#_Toc162005539)

[2 Teori 2](#_Toc162005540)

[2.1 Logistisk Regression 2](#_Toc162005541)

[2.2 K-Nearest Neighbors(KNN) 2](#_Toc162005542)

[2.3 Support Vector Machine (SVM) 3](#_Toc162005543)

[2.4 Random Forest 3](#_Toc162005544)

[3 Metod 4](#_Toc162005545)

[4 Resultat och Diskussion 5](#_Toc162005546)

[5 Slutsats 6](#_Toc162005547)

[6 Teoretiska frågor 7](#_Toc162005548)

[7 Självutvärdering 8](#_Toc162005549)

[Källförteckning 9](#_Toc162005550)

# Inledning

Denna rapport utforskar olika maskininlärningsmodeller och ser vilken som bäst passar för datasetet MNIST. Sedan år 1994 har MNIST använts inom olika områden och ses som ett grundläggande dataset för alla som sysslar med maskininlärning.

Syftet med denna undersökning var att utvärdera vilken av de fyra valda modellerna som gav bäst prediktering och samtidigt hade en rimlig beräkningstid - så frågeställningen var alltså vilken av de fyra som var den bästa modellen. De fyra modeller som användes var: Logistisk Regression, KNN, SVM samt Random Forest. Anledningen till att jag valde att avgränsa mig till dessa fyra modeller är att de alla är bra modeller när man använder just MNIST.

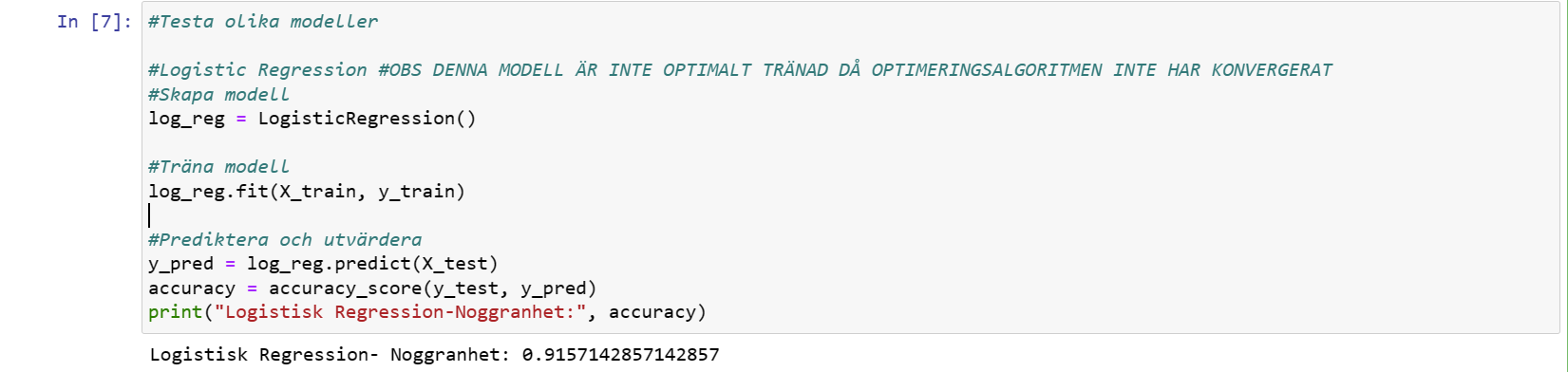
# Teori

## Logistisk Regression

Logistisk regression är en modell som ofta används för att beräkna sannolikheten att en viss data tillhör en viss klass (Géron, 2023, s.164). Den räknar ut, precis som med linjär regression, en viktad summa av datan som förs in - men till skillnad från linjär regression så beräknar denna modell det logistiska resultatet snarare än det direkta resultatet.

Ett exempel på hur en ekvation för logistisk regression kan formuleras:

#### 2.1.1 Hyperparameter

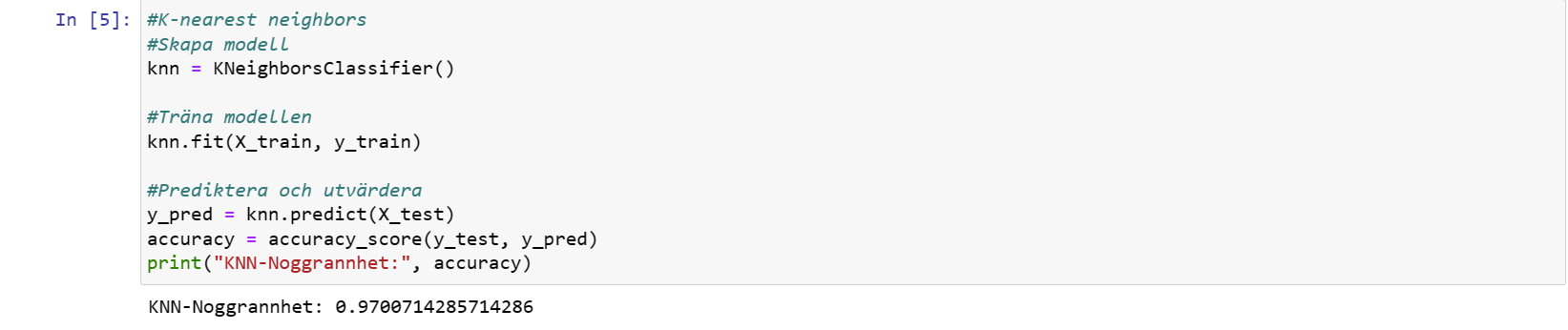
Hyperparameter testades men valdes inte då beräkningstiden blev för lång utan egentlig förbättring av resultatet. Men på grund av bristen av hyperparameter så är modellen inte optimalt tränad då optimeringsalgoritmen inte optimerades.

## K-Nearest Neighbors(KNN)

K-Nearest neighbors(KNN) är när man låter algoritmen titta på de närmaste värden till datan i fråga och sedan räknar ut ett genomsnitt på dessa data (Géron, 2023, s.26).

#### Hyperparameter

Även i detta fall så testades hyperparametrar men då inte beräkningstid blev snabbare eller prediktionen bättre så bortsåg jag från dem även här.

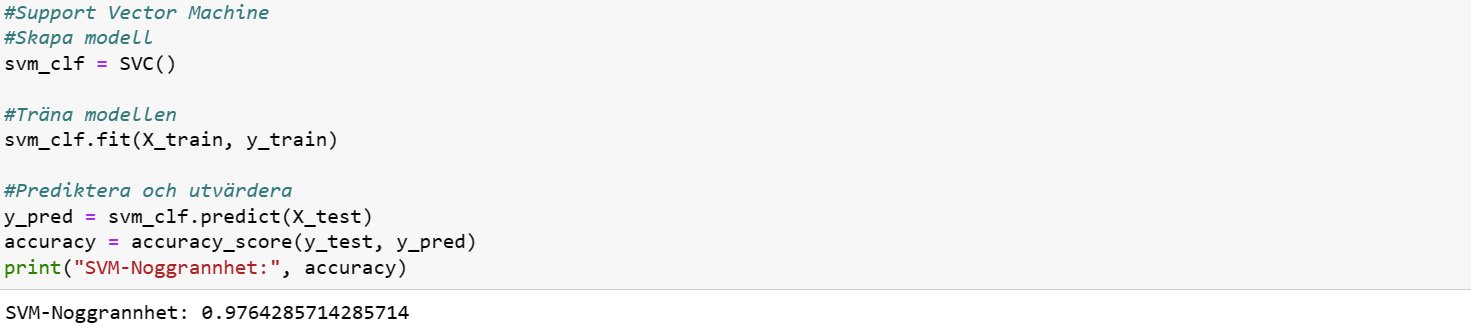


## Support Vector Machine (SVM)

Support Vector Machine är en både kraftfull och smidig modell som kan hantera både klassificering och regressionsproblem (Géron, 2023, s.175). Modellen passar bra att använda på MNIST-datan då den är särskilt effektiv på binär klassificering (exempelvis: ”är detta en fyra eller inte en fyra”).

#### Hyperparameter

Inga hyperparametrar användes (även om de ännu en gång testades) av samma skäl som ovan.

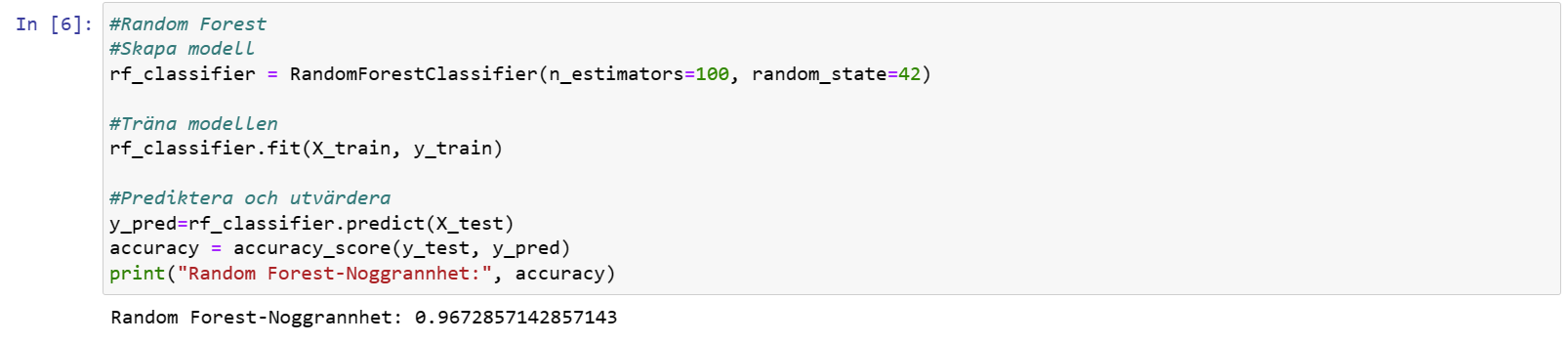


## Random Forest

Till sist prövade jag modellen Random Forest, som är en ensemble-metod som sammanställer (”ensemble”) flera beslutsträd (Géron, 2023, s.211). Random Forest algoritmen för in extra slumpmässighet (”randomness”) när beslutsträden utvecklas och algoritmen får en större mångfald vilket ger en generellt sett bättre modell (Géron, 2023, s.220). Den fungerar också bra för både klassificering och regression.

#### Hyperparameter

De enda hyperparametrarna som användes var de som är inbyggda i modellen såsom n\_estimators, criterion, max\_depth med flera.



# Metod

Datan MNIST hämtades från uppgiftsbeskrivningen som givits till denna uppgift. Miljön som användes för programmeringen var Jupyter Notebook där hjälp med koden kom från boken *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow. Concepts, Tools, and Techniques to Build Intelligent Systems* av Aurélien Géron (2023, Third Edition. O’Reilly, Sebastopol) tillsammans med programmeringsstöd från ChatGPT.

En bild som visar text, skärmbild, Teckensnitt

Automatiskt genererad beskrivningFörst laddade jag ner datan och därefter ville jag få en överblick över den och se om det fanns några saknade värden, och i så fall åtgärda dessa. Jag kunde gå vidare snabbt då det inte fanns några saknade värden. Efter det började jag utvärdera de olika modellerna jag valt och även potentiella hyperparametrar. Efter att ha testat mig fram så kom jag fram till att ingen av de modeller jag använde nödvändigtvis drog fördel av att ha en hyperparameter. Beräkningstiden blev lång och prediktionerna blev inte signifikant bättre av att använde hyperparametrar. Jag delade upp datan i tränings- och test-sets och använde noggrannhet (accuracy) som prestandamått då den fungerar väl för klassificeringsmodeller. En modell som jag prövade tillsammans med hyperparametrar var Gradient Boosting Machines (GBM) men då den tog mycket lång tid på sig så valde jag att inte fortsätta med den modellen. Se bild:

# Resultat och Diskussion

Resultatet av studien visade att alla fyra modeller jag prövade gav bra resultat (se tabell) då de alla predikterade med en säkerhet på över 90 %. Men som framgår av tabellen var den bästa modellen Support Vector Machine som predikterade en noggrannhet på 0.9764285714285714. Någonting att ha i åtanke är att resultaten kan variera något när man kör det fler gånger och då kan en annan modell visa högre prediktion. Detta kan exempelvis bero på slumpmässighet i algoritmen och/eller i datan.

|  |  |
| --- | --- |
| Modell | Prediktion |
| Logistisk regression | 0.92 |
| KNN | 0.97 |
| SVM | 0.98 |
| Random Forest | 0.97 |

Tabell över modellernas noggrannhet avrundat till två decimaler.

Det är tydligt att alla modellerna fungerar väl men förbättringar som kan göras hade kunnat vara att ha fler modeller och ha testat fler hyperparametrar för ytterligare finjustering. Man kan alltid också köra modellerna fler gånger för att utvärdera eventuella förändringar. Men efter övervägande av tid och kvalité på prediktionerna så anser jag resultaten så pass goda att det inte är motiverat att lägga mer tid i detta skede.

# Slutsats

Min frågeställning i denna rapport har varit att besvara vilken, av de fyra modellerna jag valde, som fungerade bäst när datan MNIST användes. De fyra prövade modellerna var Logistisk regression, K-nearest neighbor, Support Vector Machine och Random Forest. Alla fyra fungerade väl men SVM fungerade allra bäst. Min slutsats är att alla fyra modellerna gav så pass bra resultat, med en prediktion på över 90%, att de kan alla användas men att SVM ändå är den som är mest optimal.

# Teoretiska frågor

1. Du tar en del av datan för att träna modellen på, sen använder du testdelen för att se hur bra modellen fungerar på ny data. Validering är när man tränar flera modeller med olika hyperparametrar på hela setet minus valideringssetet (det så kallade reducerade träningssetet).

2. Det Julia kan göra är att istället dela upp datan i många små valideringsset och att då utföra upprepade kors-valideringar med dessa små set – detta kallas cross-validation.

3. Ett regressionsproblem är när man har beroende variabler som förutsäger ett resultat. Exempelvis att förutsäga vad ett hus kan säljas för beroende på läge, storlek eller skick eller vilken lön en person har beroende på utbildning och ålder (och kön sorgligt nog). Exempel på regressionsmodeller är linjär regression, polynomial regression och Random Forest-regression.

4. RMSE, som står för *Root Mean Square Error*, är ett vanligt performance measure för regressionsproblem. Den visar på hur mycket fel systemet gör i sina prediktioner (hur mycket fel den typiskt gör) och den lägger större vikt vid stora fel.

5. Klassificeringsproblem är när vi har två möjliga värden (exempelvis ”churn” eller ”ej churn). Några exempel på modeller är även här linjär regression (även om det snarare är linjär klassificering), Support Vector Machine och Random Forest Classification. Klassificering kan användas för att bestämma om en bild föreställer en blomma eller inte eller för att känna igen handskrivna siffror (mnist). Confusion Matrix är en matris som visar på fyra möjliga resultat av när man predikterar något. Falsk negativ, Falsk positiv, Sann positiv och Sann negativ. Lättast förklarast med gravidexemplet: Falsk negativ – du förutsäger att en gravid person inte är gravid. Falsk positiv – du säger att en icke-gravid är gravid när den inte är det. Sann positiv – du säger att en gravid person är gravid. Sann negativ – du säger till en icke-gravid person att den inte är gravid.

6. K-means modellen är en algoritm, som ibland kallas Lloyd-Forgy algoritmen efter de två upphovsmännen, som slumpmässigt lokaliserar kluster i den givan datan. Genom hyperparametern n\_clusters kontrollerar den hur många kluster den letar efter.

7. Ordinal Encoding är en klass som låter en konvertera olika kategorier till numeriska värden och den fungerar bra när de ordnade kategorierna är på en skala ex. från väldigt dåligt, dåligt, OK, bra, väldigt bra. One-Hot Encoding gör samma sak men skillnaden är att det inte finns någon skala här utan man letar efter en viktig egenskap (hot) medan alla andra värden inte är det (cold). Exempelvis om ett hotellrum har havsutsikt eller inte. Dummy Variabel Encoding är som One-Hot Encoding men i det här fallet så fortsätter vi med den binära uppdelningen. Exempelvis om vi tittar på utbildningsnivån i ett land kan vi först titta på de som gått klart grundskolan vs de som inte gjort det – sen fortsätter analysen med de som gått ut grundskolan och tittar på de som gått ut gymnasiet vs de som inte gjort det osv.

8. Jag tänker att det är Julia som har rätt eftersom vi antingen kan bestämma att datan både kan delas in i separata diskreta kategorier(röd, grön, blå) men eftersom färg är ett spektrum så tolkar vi vilken kategori en viss färg tillhör(är färgen ”petrol” blå eller grön?).

9. Streamlit är en ”open-source” applikation där du med python kan skapa webb-appar med hjälp av maskininlärning och data science. I och med att det är open-source kan alla använda den och den är lätt att dela med andra.

# Självutvärdering

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

Svar: Hyperparametrar var något som var utmanande. Både i den aspekten om vilken hyperparametrar som är de bästa av olika anledningar till modellen i fråga – men också i den aspekten om modellen man jobbade med överhuvudtaget behöver en hyperparameter och potentiella risker som kan medfölja om man väljer bort det. Att väga för- och nackdelar och tänka ut vad som är bästa lösningen. Det var något som var mer krävande och tog mer tid än jag trodde.

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

Svar: Jag hoppas att mitt arbete visar att jag förstått och även visar att jag kan jobba med maskininlärning på en grundläggande och godkänd nivå.

1. Något du vill lyfta fram till Antonio?

Svar: Utmanande men rolig kurs och tydligt förklarande över annars svåra områden inom maskininlärning.

# Källförteckning

ChatGPT (2024-03).

Géron, A. (2023) *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow. Concepts, Tools, and Techniques to Build Intelligent Systems*. Third Edition. O’Reilly, Sebastopol.

Prgomet, A.(2024) Föreläsningar vid EC Utbildning.