MNIST

Rapport över modelleringen av MNIST-data

Introduktion

Bakgrund:
Att programmera en dator för att identifiera mönster i data och sedan göra förutsägelser är känt som maskininlärning. MNIST-databasen med handskrivna siffror är ett välkänt dataset som används flitigt inom maskininlärning och används när man vill utvärdera och testa modeller.
Frågeställning:
Syftet med denna rapport är att få kunskap i bästa ackurata utfall den bästa modellen kan tillföra.
Min vetenskapliga fråga lyder:
“Predikterar en mer eller mindre flexibel modell bättre på 10 % av MNIST-datasetet?”

Databeskrivning

Datasetet MNIST är skapat av Yann LeCun, Corinna Cortes, Christopher J.C. Burges, innehållandes 70 000 bilder på, av amerikanska studenter, handskrivna siffror från 0 till 9. Därmed finns 10 klasser av de olika siffrorna.
Datan som används i denna rapport kommer att vara 10% (7000 variabler) av hela datasetet. 80% används som träningsdata, 20% används till trestdata. Vi har då 5600 rader och 784 kolumner med träningsdata och 1400 rader med 784 kolumner av testdata.
Varje rad representerar en handskriven siffra och kolumnerna visar ett nummer mellan 0 - 255 där 0 är vitt och 255 är svart med en storlek på 28x28 pixlar vardera. Om man tittar på pixlarna i koden kommer kanterna att visa 0 vilket är vitt.

Metod & modeller

Jag jämförde fyra modeller och jämförde dess ackurata prediktionsförmåga. De modeller som används är logistic regression, RandomForest, K-NearNeighbor samt Support Vector Machine. Den sistnämnda använde jag gridsearch på för att se om det blev en stor skillnad på denna och de andra modeller där hyperparametrar ställs in manuellt.

Random Forest Maskininlärningsalgoritmen random forest kombinerar resultatet från olika beslutsträd för att producera ett enda resultat. Dess kan lösa både klassificerings- och regressionsproblem. Beslutsträd är uppbyggt av noder och grenar. Vid rotnoden beslutas om en predikterad datapunkt tillhör ett klass eller värde varpå den predikerade datapunkten placeras i tillhörande nod nedaför. Denna process upprepas tills den nått en lövnod. Lövnoden är “ren” alltså att det bara finns ett alternativ, att den inte delar sannolikhet med ett annat värde.
K-NearNeighbor Vid ett klassificeringsproblem, gör KNN att vid en prediktion tas den närmaste tränings-datapunkterna från denna förutspådda datapunktens position. Med hjälp at hyperparameter kan vi bestämma hur många röstande punkter som skall användas för att rösta fram vilken kategori den nya datapunkten ska tillhöra.
Logistic Regression Prediktiv analys och klassificering använder sig ofta av denna typ av statistisk modell. Logistisk regression beräknar sannolikheten för att en händelse inträffar. Med tanke på att

resultatet är en sannolikhet är den beroende variabelns intervall 0 till 1. Vid logistisk regression transformeras sannolikheten för framgång dividerad med sannolikheten för att misslyckas.
SVM som står för Support Vector Machine är en övervakad inlärningsmodell och analyserar både klassificering och regressionsanalys. Den fungerar som så att på träningsdatan markeras datapunkter som tillhör de olika kategorierna och som kommer att fungera som en markering för vart gränsen går för att en testdatapunkt ska tillhöra denna klass eller inte. Helst ska gapet mellan de olika gränsvecktorerna vara så bred som möjligt.

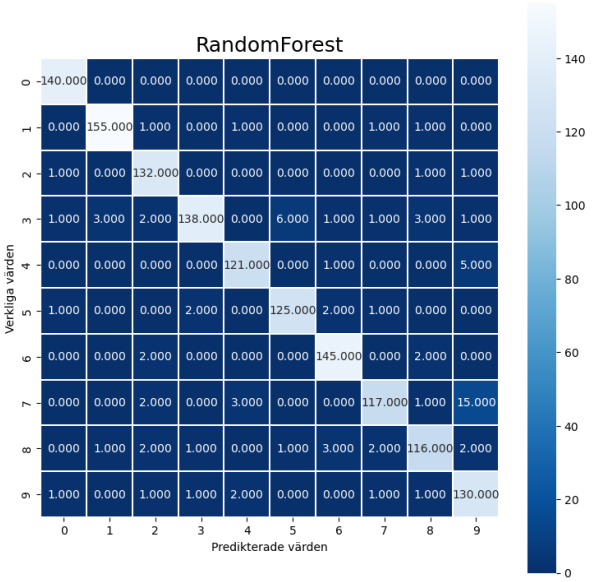
Projektets resultat och analys

Resultat:
Enligt tabellen kan vi se att alla modeller gav tillfredställande resultat men RandomForestClassifier fick en något högre score än SVM och K-NearNeighbor där SVM var något bättre än RandomForest. LogisticRegression gav dock ett något sämre utfall än de andra tre.

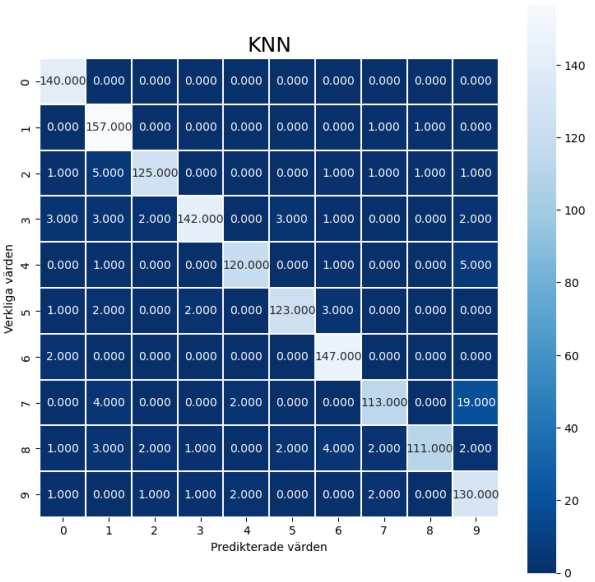
Modell	Precision
RandomForestClassifier	0.9507142857142857
K-NearNeighbor	0.9342857142857143
LogisticRegression	0.8928571428571429
SupportVectorMachine	0.94

Analys:
RandomForest hyperparametrar ställdes in så att n_estimators låg på 200 max_depth låg på 17 och bootstrap=False. Genom att justera dessa åstadkom marginella förbättringar. Även om max_depth raderades blev det bättre med nivå 17 på denna hyperparameter.
För K-Near Neighbours hyperparameter valdes först 10, vilket gav sämre resultat än när 6 testades. Fler röstande datapunkter gav alltså ett sämre resultat vilket påvisar att datan kan vara känslig för övergång mellan klasserna.
Varför RandomForest fick högsta noggrannhetsprestanda kan bero på att den är mer flexibel än LogisticRegression. LogisticRegression och andra sidan verkar vara en enklare modell än RandomForest eftersom den på en binär nivå endast väger sannolikheten mellan klasserna.
Då en modell är en matematisk representation av en process som används för att prediktera en uppsättning data, finns grader av flexibilitet hos olika modeller och vad som gör en modell flexibel är modellens förmåga att förändra, utvecklas och lära av data.
Support Vector Machine ställdes in med hyperparametrar med GridSearchCV. Modellen verkar vara effektiv för flexibla klassificeringsproblem med mycket olinjära beslutsgränser så som i mnist. Denna verkar vara ett enkel och okomplicerad sätt att arbeta och få fram en bra modell.

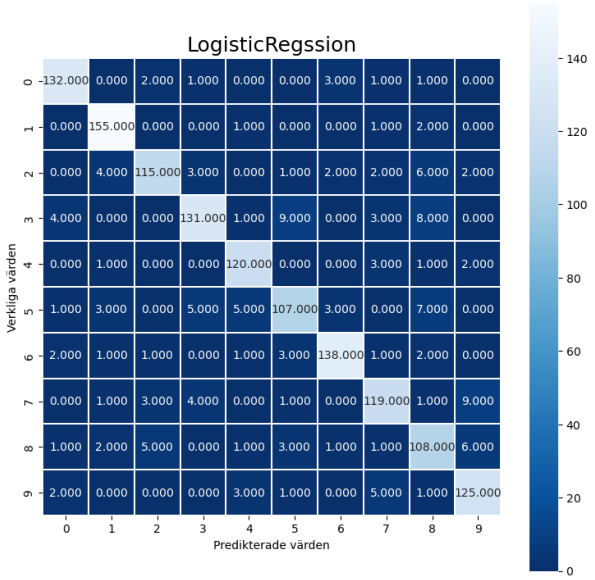
Confusion Matrix



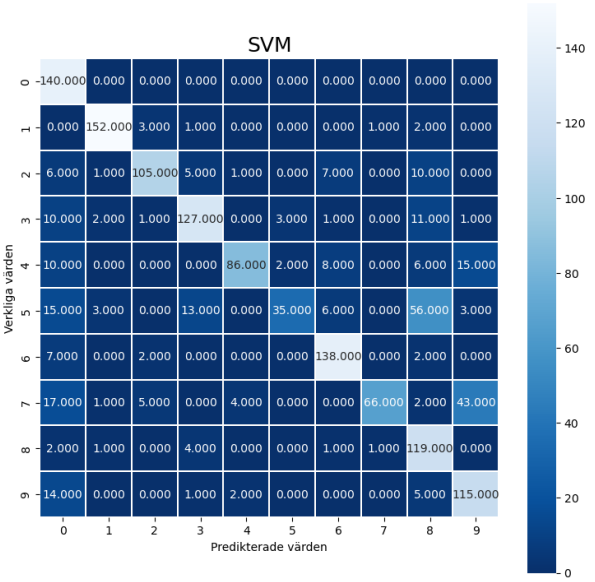
Random Forest, modellen med högsta score. Det verkliga värdet “9” predikteras rätt 130 gånger men predikteras fel totalt 24 gånger, vilket är 15,58%. Vad som är intressant är att 9 förväxlades med 7, totalt 15 gånger, vilket är 9,74%.



K-Near Neighbor, modellen med näst högstascore. Här förväxlas 9 med 7, 19 gånger vilket är 11,95 %



LogisticRegression, modellen med sämst score. Här förväxlas 9 med 7 på testdatan vilket är endast 9 gånger vilket är 6,25% men denna modell hade över lag svårare att särskilja på andra klasser som till exempel 5 och 3. Detta medför att modellen är lite mindre stabil än de andra modellerna.



Trots att denna modell ställdes in med gridsearch blev den endast näst bäst med 94% träffsäkerhep på prediktionerna. Denna gav många fel på några få siffror medan tex alla sanna värden med siffran 0 predikteras fläckfritt.

Vad som kan tänkas vara intressant är RandomForest förväxlade 9 med 7, totalt 15 gånger, vilket är 9,74%. Jämför man det med LogisticRegression som förväxlades 9 med 7 endast med 6,25%. Det är ett bättre utfall på just den prediktion men modellen hade över lag svårare att särskilja på andra klasser som till exempel 5 och 3. Detta medför att modellen är lite mindre stabil än de andra modellerna.

Slutsats och förslag på potentiell vidareutveckling:

Genom att jämföra fyra olika modeller (RandomForest, KNN och LogisticRegression samt SVM) blev resultatet att en mer flexiblel modell som Randomforest gav en högre korrekt noggrannhet än en modell med lägre flexibilitet samt en enklare modell inte gav bättre resultat än en mer avancerad.

Som ett nästa steg i utvecklingen skulle fler modeller kunna användas samt ta en större del av datasetet. Man skulle även kunna använda sig av cross validation som använder olika datamängder för att testa och träna modeller över ett givet antal iterationer. En Voting Classifier hade också kunnat användas för att få reda på den bäst predikterande modellen.