Modellering av MNIST data

Introduktion

- Bakgrund

Maskininlärning används för att träna avancerade matematiska modeller med data för att kunna få svar på viktiga frågor. Den används i situationer där uppgiften eller data ständigt förändras. Man vill oftast utföra prediktioner från data därför måste man identifiera mönster och träna modeller tills man hittar den bästa med högst prediktionsförmåga.
Maskininlärning handlar lära en dator att utföra uppgifter utan direkta instruktioner. Det betraktas som en del av artificiella intelligensen.
Beroende av data man har kan maskininlärning delas i Supervised and Unsupervised learning där i Supervised learning data har etiketter vilket gör att man kan hantera den data lättare. I Unsupervised learning är data utan ”labels” och där vill man lära datorn att hitta mönster och relationer genom att gruppera data i kluster.

- Syfte och frågeställning

Syftet med denna uppgift är att träna olika modeller och välja den med bäst prediktionsförmåga. Vi kommer modellera det kända datatestet MNIST som innehåller handskrivna siffror från 1 till 9. Målet är att hitta den bästa modellen som kan korrekt identifiera största antal siffror. Resultatet kommer att mätas med Accuracy score vilket är ett vanligt utvärderingsmått för en klassificieringsproblem. Den tar antal korrekt predikterade observationer delat med antal observationer som från testsetet. Den är enkel och passar bra för detta klassificeringsproblem som jag vill lösa. Sedan kommer resultatet plottas med Confusion Matrix för att visualisera resultatet.

Databeskrivning/EDA (Exploratory Data Analysis)

MNIST dataset består 70 000 bilder på handskrivna siffror splittrad till träningsdata med 60 000 och testdata med 10 000 testbilder. Detta är alltså rader. Varje rad är en bild med bild med 28x28 pixlar vilket är 784 pixlar per bilden. Varje pixels intensitet mäts på skalan från 0 till 255 där 0 är helt svart och 255 helt vitt. Enligt skalan kan algoritmerna lära sig att identifiera siffrorna på bilder.

Detta dataset används ofta för att lära sig och öva på maskininlärning genom att testa olika modeller för att uppnå högsta prediktion.

KLISTRA IN BILDER

Metod och Modeller

Första modellen som användes var *Support Vector Klassificerare* 'SVC' för att den är bra på att hantera högdimesionella data som MNIST datasetet. SVC ritar linje som separerar data i olika klasser. Därefter fittas modellen med all träningsdata och använder Grid Search för att hitta de bästa hyperparametrarna. **Kernel** hade två val "rbf" och "poly". Båda kernels används för att hantera icke linjära data. Resultaten var mätt med Accuracy och Confusion Matrix

Nästa modell som tränades *var Logistisk Regression* och det är en binär klassificerare som uppskattar sannolikheten att en observation tillhör en klass. Modellen använde Gridsearch där justerade hyperparametrarna var **Penalty** med None, *"*l1*"* och "l1*"* straffar. L1 straffar genom att kvadrera koefficienten i förlustfunktionen. Därefter **Solver** med "lbfgs", "liblinear" och "sag" där 30 kombinationer misslyckades för att inte alla hyperparametrar passar med varandra. Därefter fittades modellen med dimensionell reducerad och skalade testdata och resultatet var mätt med Accuracy och Confusion Matrix

Tredje modellen som användes var *Random Forest* som består av flera beslutsträd som tränas på slumpmässiga träningsdata. Denna modell fittade datan strax efter splittring och då uppnådde högst resultat enligt Accuracy. Hyperparametrar vi testade var **n_estimators**, **max_depth** och **min_samples_split**. Resultatet var mätt med Accuracy och Confusion Matrix

Projekt Resultat och Analys

Resultat

	Modeller	Accuracy med mindre data	Accuracy med all data	Transformering/Tuning
1	SVC	0.95	0.98	Skalad, GridSearch
	SVC (dim red)	0.87	inte testat	Reducerad dimension, Skalad, GridSearch
2	Logistic regression	0.89	0.91	Skalad
	Logistic regression (dim red)	inte testat	0.92	Reducerad dimension Skalad, GridSearch
3	Random Forest	0.95	0.95	Reducerad dimension, GridSearch

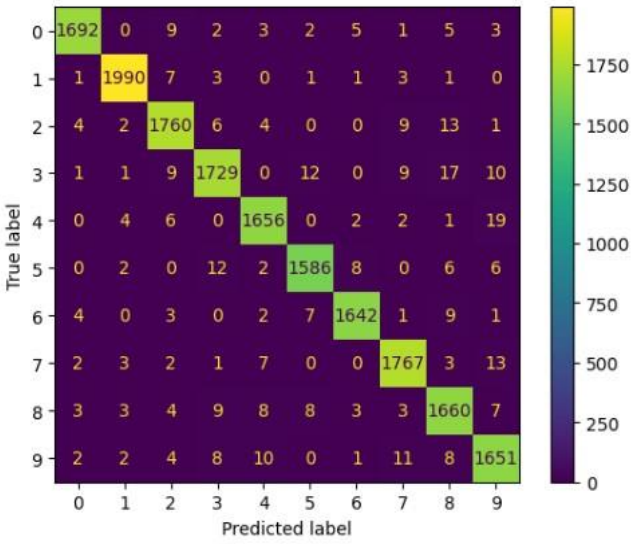
Analys

Som vi kan se i tabellen så bästa prediktionsförmåga hade Support Vector Klassificerare med en utvärdering på 98% med Accuracy mätaren. SVC modellen brukar inte rekommenderas för dataset med tiotusentals kolumner. MNIST är precis på gränsen därför tog det först 10 timmar

innan jag började justera data genom att använda dimensionell reducering för att snabba upp processen. Detta gjordes med PCA där vi behåll 95 % varians och få ner antal dimensioner från 784 till 154.

För att hitta de bästa hyperparametrarna så fittade jag modellen först med mindre data, 5 000 observationer som träningsdata och 1 000 som testdata för att snabba upp processen. Datan var normaliserad med Standard Scaler och Randomized search letade efter den optimala kombinationen av hyperparametrar med Cross Validation principen. **Gamma** justerade olika värden som hade en reciprocal fördelning vilket innebär att högre siffror är mer troliga. **C** hyperparameter justerade värden med jämn fördelning. De två hyperparametrar påverkar modellens flexibilitet. Med **best_params_attributen** kunde man komma åt de bästa hyperparametrarna som jag stoppade in i Grid Search för att testa alla kombinationer. Processen upprepades fast med en ny hyperparameter **kernel.** Den hade två val, ’rbf’ och ’polly’. Polly är polynomisk kernel som hanterar icke linjära samband i datan som blev utvald av Grid Search.

Nu upprepade vi processen med all data splittrad i train och testdata. Antal dimensioner var reducerad till 154 med PCA. Vi använde Grid Search återigen med polynomisk **kernel** ”polly” eftersom den visade sig effektiv på mindre data vilket innebär att den hantera bra den typen av data. **Gamma** och **C** var justerade för att se hur de påverkar modellen med mer data. Grid Search tog inte så lång tid att hitta bästa hyperparametrar så jag fittade modellen med ej dimensionella reducerade träningsdata och justerade **gamma** och **C** och svaret blev utmärkt. Accuracy score har mätt 98% modellens prediktionsförmåga.



Som vi kan se på bilden har modellen gjort största misstag med siffran 4 som tolkades som 9 för nitton gånger. Sedan tolkade modellen treor som åttor många gånger men skillnaden mellan dessa handskrivna siffror kan ibland bara upptäckas med ögat

Slutsats och förslag på potentiell vidareutveckling

När det kommer till modellerna som har använts i denna uppgift så hade man kunnat justera hyperparametrarna ännu mer för prestandaförbättring men detta är tidskrävande och resultatet på 98% Accuracy som SVC modellen har uppnått kändes tillfredställande.
Det finns många andra modeller som löser klassifikationsproblem såsom Linear Regression modell, Decision Trees, Neural networks och Voting Classifier som kan skulle kunna modellera MNIST data. Dessutom kan få av dem slå nuvarande resultatet. Voting Classifier kan visa sig effektivare men det är svårt att säga mycket innan man tränar modellen. Den klassificierare som kan potentiellt slå resultatet med rätt modellering är Neural network och detta kommer att undersökas i nästa kurs.