Handwritten Digit Recognition Using Machine Learning

Ali Khalil

EC Utbildning

Maskininlärning

2024–03

Abstract

This project focuses on handwritten digit recognition using a machine learning model trained on the MNIST dataset. The images are preprocessed through grayscale conversion, binarization, resizing, contour detection and normalization to match the model's input format. A pre-trained XGBoost classifier is used for digit classification. While the final model achieves high accuracy, the process was not without challenges, including optimizing preprocessing techniques. Nevertheless, the results demonstrate the effectiveness of machine learning for image classification, and future improvements could include data augmentation and deep learning models for better performance.

Erkännanden

Jag vill tacka mina lärare för deras vägledning under maskininlärningskursen, och särskilt Antonio för hans stöd. Jag uppskattar verkligen hur han förklarar komplexa ämnen och gör dem lättare att förstå. Hans undervisningsförmåga, tillsammans med hans tålamod, har varit ovärderlig.

1 Innehållsförteckning

2 Inledning

Maskininlärning har under de senaste åren blivit ett allt viktigare verktyg inom många områden, från hälsovård och finans till transport och e-handel. Genom att analysera stora mängder data kan maskininlärningsmodeller identifiera mönster och fatta beslut utan direkt mänsklig inblandning. Denna teknik har revolutionerat många industriella och kommersiella tillämpningar och skapat nya möjligheter för effektivisering och innovation.

I denna rapport fokuseras på en specifik aspekt av maskininlärning, nämligen användningen av XGBoost-modellen för handskrivna siffrors igenkänning, baserat på MNIST-databasen.

2.1 Syfte

Syftet med denna rapport är att undersöka och implementera en maskininlärningsmodell för att upptäcka handskrivna siffror ges av användaren på olika sett, antingen genom all ladda upp en bild, använda kameran eller skriva siffran direkt i appen .Modellen tränades med MNIST-datasetet som innehåller 70 000 gråskalebilder av siffror (0–9), målet med detta projekt är att bygga en modell som kan klassificera siffror med hög noggrannhet

2.2 Frågeställning

1. Hur effektiv är maskininlärning vid igenkänning av handskrivna siffror, och vilka är dess begränsningar?

2. Hur kan den valda modellen förbättras i termer av noggrannhet, effektivitet och generalisering, och vilka alternativa metoder kunde ha använts?

3 Teori

Denna sektion presenterar teori som är relevant för att förstå kontexten för detta projekt, modellerna och de utvärderingsmått vi använde

För vår MNIST-klassificering använde vi Beslutsträd, Random Forest och XGBoost, som alla är träd-baserade modeller kända för sin effektivitet och tolkbarhet i strukturerade datauppgifter.

3.1 Beslutsträd

Ett beslutsträd är en icke-parametrisk övervakad inlärningsalgoritm som används för både klassificerings- och regressionsuppgifter. Det har en hierarkisk trädstruktur, som består av en rot-nod, grenar, interna noder och blad-noder. (IBM, u.d.)

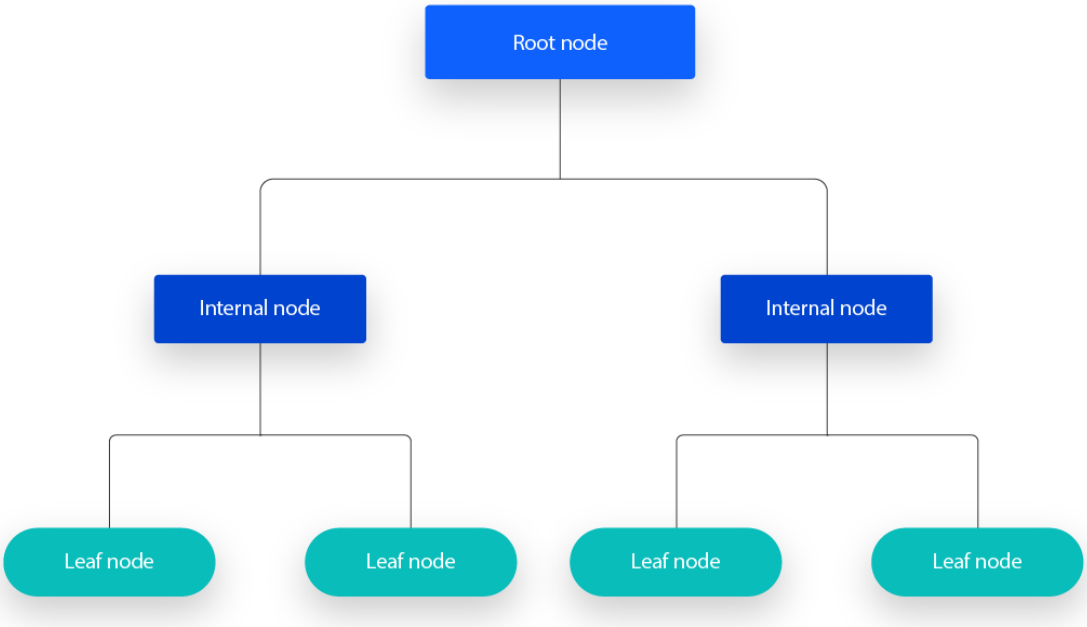

Figure 1 :https://www.ibm.com/think/topics/decision-trees

3.2 Random forest

Random forest är en ensembleteknik som kombinerar resultatet från flera beslutsträd till ett enda resultat. Den aggregerar förutsägelsen från alla träden och väljer sedan det mest populära svaret som det slutgiltiga resultatet. Varje träd överväger en slumpmässig delmängd av dataegenskaper, vilket hjälper till att säkerställa låg korrelation mellan träden. (IBM, u.d.)

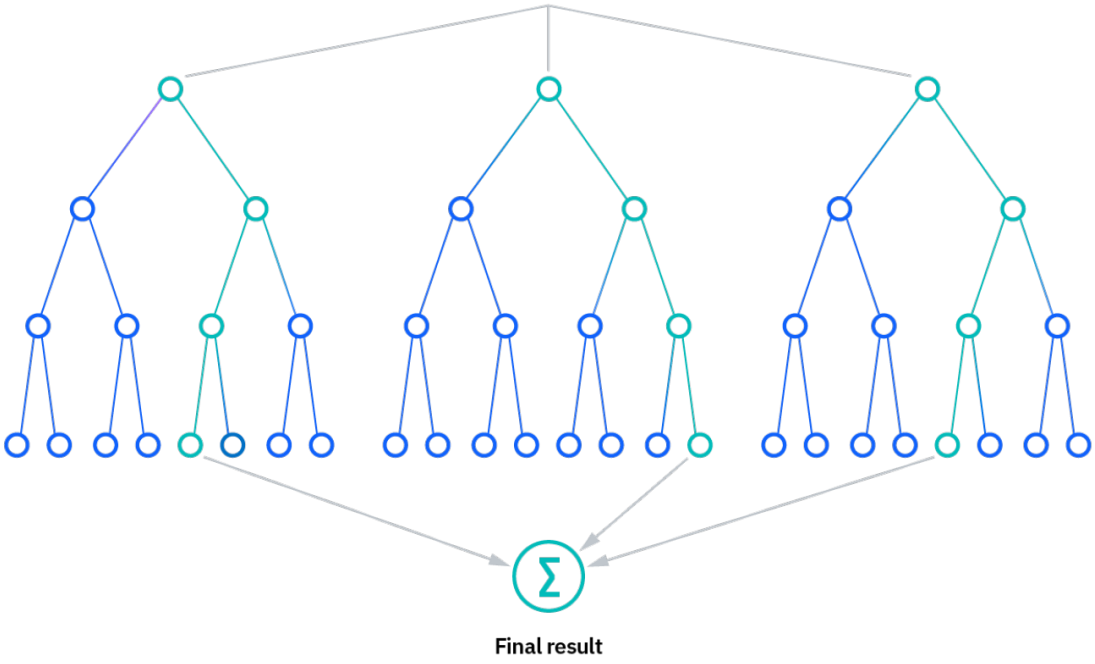

Figure 2 : https://www.ibm.com/think/topics/classification-machine-learning

3.3 XGBoost

Gradientboostade beslutsträd är en typ av boosting-algoritm som använder gradientnedstigning. Liksom andra boosting-metoder, börjar gradientboosting med en svag lärandealgoritm för att göra förutsägelser. Det första beslutsträdet i gradientboosting kallas för basläraren. Därefter skapas nya träd på ett additivt sätt baserat på baslärarens misstag. Algoritmen beräknar sedan residualerna för varje träds förutsägelser för att avgöra hur långt modellen var från verkligheten. Residualer är skillnaden mellan modellens förutsagda och faktiska värden. Residualerna sammanställs sedan för att bedöma modellen med en förlustfunktion. (IBM, u.d.)

Figure 3

3.4 Utvärderingsmått

För att bedöma modellens prestanda använde vi följande klassificeringsmått:

3.4.1 Acuuracy :

Mäts som andelen korrekt klassificerade siffror.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

3.4.2 Precision:

Precision är en måttstock som mäter noggrannheten av positiva förutsägelser och betonar förhållandet mellan korrekt förutsagda positiva observationer och det totala antalet förutsagda positiva. (Géron, 2019, s. 91)

$$\text{Precision} = \frac{TP}{TP + FP}$$

3.4.3 Recall (Sensitivitet):

Även känd som sann positiv rate, handlar det om att fånga andelen korrekt förutsagda positiva observationer i relation till alla observationer i den faktiska positiva klassen. (Géron, 2019, s. 91)

$$\text{Recall} = \frac{TP}{TP + FN}$$

3.4.4 F1-score:

Ett harmoniskt medelvärde av precision och recall, som balanserar falska positiva och falska negativa. (Géron, 2019, s. 92)

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

där:

- TP = True Positives
- TN = True Negatives
- FP = False Positives
- FN = False Negatives

4 Metod

I denna sektion beskriver vi ursprunget och förberedelserna av datan. Vi förklarar också hur vi valde lämpliga maskininlärningsmodeller, samt hur vi utvärderade deras prestanda. Vidare går vi igenom de experiment vi genomfört för att träna och testa modellerna.

4.1 Tools

Vi använde Python för att implementera maskininlärningsmodeller och hantera dataanalysuppgifter.

NumPy: Användes för effektiv hantering av numeriska data och arrayoperationer.

Pandas: Användes för datamanipulering, rengöring och analys.

Matplotlib/Seaborn: Dessa bibliotek användes för att visualisera data och plottar resultaten från våra modeller.

scikit-learn: Användes för att implementera maskininlärningsmodeller som besluts träd och random forests, samt för att utvärdera modellernas prestanda.

XGBoost: Användes för att implementera en kraftfull boostad algoritm för klassificering.

Pickle: Användes för att spara och ladda modeller efter träning.

cv2: Användes för bildbehandling

Streamlit: Användes för att bygga en användarvänlig webbapp för att visa och interagera med

4.2 Data:

Den data som användes i detta projekt var MNIST-datasetet (Modified National Institute of Standards and Technology), som är ett av de mest använda datasetten inom maskininlärning, särskilt för bildklassificering. Datasetet består av 60 000 träningsbilder och 10 000 testbilder, där varje bild är en 28x28 pixel svartvit bild som föreställer en handskriven siffra (0–9). Varje bild är märkt med den korrekta siffran, vilket gör det möjligt att använda datasetet för att träna och utvärdera klassificeringsmodeller.

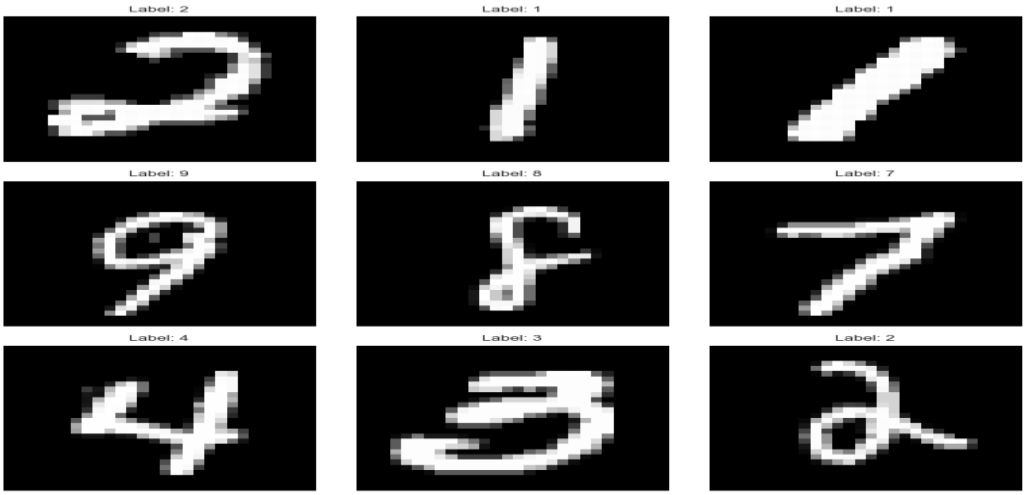

Figure 4

4.2.1 Distribution of digits

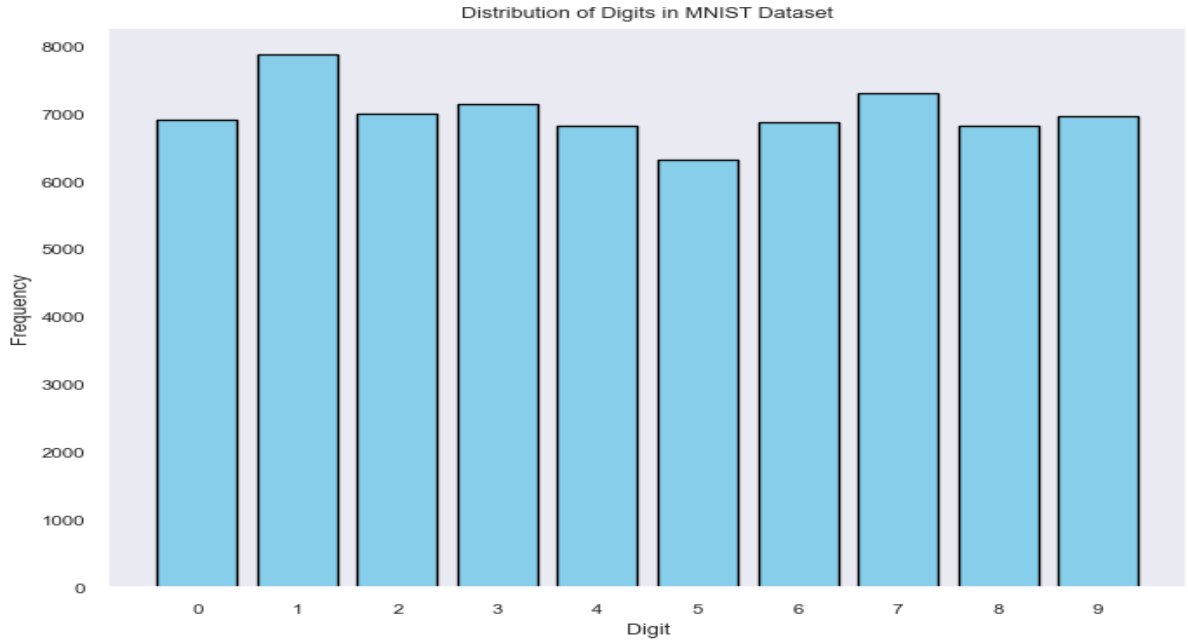

Figure 5

4.3 Träning och testning

För att träna och utvärdera våra modeller genomfördes följande steg:

4.3.1 Normalisering av data

För att säkerställa att alla pixelvärden ligger inom samma skala normaliserades bilddata genom att dividera med 255. Detta konverterar pixelvärdena från intervallet [0, 255] till [0,1], vilket gör att modellerna kan tränas mer effektivt.

4.3.2 Dataluppdelning

Först delades datasetet upp i en träningsmängd (80 %) och en testmängd (20 %). Testmängden används endast för slutlig utvärdering.

Därefter delades träningsmängden ytterligare i en mindre träningsmängd (80 % av träningsdatan) och en valideringsmängd (20 % av träningsdatan). Valideringsmängden används för att utvärdera modeller under träning och justera hyperparametrar.

4.3.3 Tre klassificeringsmodeller testades:

Random Forest, Beslutsträd (Decision Tree), XGBoost

För att säkerställa att data var korrekt skalad innan träning användes en StandardScaler i varje pipeline, vilket standardiserar data genom att subtrahera medelvärdet och dividera med standardavvikelsen.

De tre modellerna tränades på den mindre träningsmängden (X_train_small och y_train_small) istället för hela träningsdatan, för att underlätta experiment och jämförelser.

5 Resultat och Diskussion

Efter att modellerna tränats utvärderades deras prestanda och den bästa modellen finjusterades med hyperparameteroptimering.

5.1 Modellutvärdering och hyperparameteroptimering

För att mäta prestandan av varje modell testades de på valideringsmängden (X_val, y_val).

Följande mått användes för att jämföra modellerna:

Noggrannhet (Accuracy) – Andelen korrekt klassificerade siffror.

Konfusionsmatris – En visuell representation av modellens felklassificeringar.

Klassificeringsrapport – Innehåller precision, recall och F1-score för varje klass.

Modell	Accuracy
Random Forest	0,9663
Decision Tree	0,8640
XGBoost	0,9762

5.1.1 Hyperparameteroptimering av XGBoost

Eftersom XGBoost visade lovande resultat, finjusterades dess hyperparametrar med RandomizedSearchCV användes med 3-faldig korsvalidering (cv=3) och 20 iterationer för att hitta den den bästa kombinationen av parametrar.

Den optimerade XGBoost-modellen testades på testdatan (X_test, y_test) för att beräkna dess slutliga noggrannhet.

Test Accuracy of Tuned XGBoost Model: 0.9726

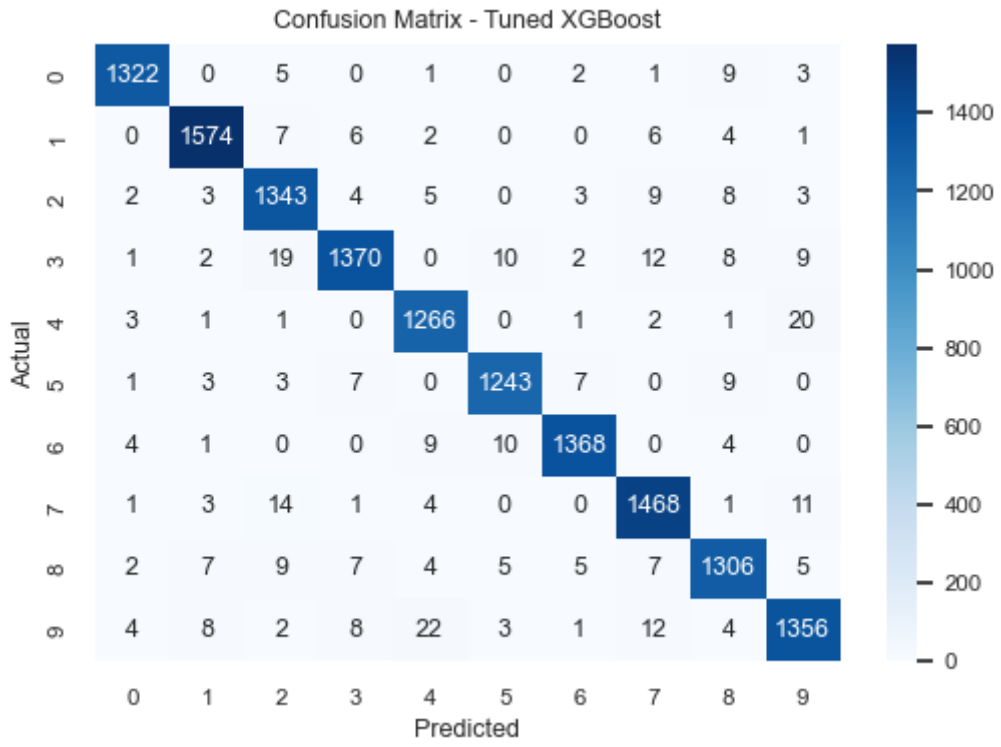

Figure 6

I slutet tycker jag att modellen fungerade bra även att jag vet att vi kan få bättre resultat med olika metoder som vi kommer att prata om i nästa avsnitt

Hög övergripande noggrannhet De flesta är korrekta, vilket tyder på stark generalisering.

Minimal felklassificering för vissa siffror som 4, 7 och 9 har mer felklassificeringar detta kan bero på likheter i handskrivna varianter.

6 Streamlit

Den här streamlit appen är designad för att känna igen handskrivna siffror med hjälp av en förtränad modell

6.1 Import och Modellinläsning:

En förtränad XGBoost-modell (xgboost_digit_classifier.pkl) laddas från en fil med hjälp av pickle för siffraklassificering.

6.2 Bildförbehandling:

Funktionen preprocess_image(image) bearbetar den uppladdade eller ritade bilden i flera steg:

Omvandlar bilden till gråskala.

Eventuellt inverterar färgerna (om det behövs för bättre kontrast).

Tillämpas ett binärt tröskelvärde för att separera siffran från bakgrunden.

Hittar konturer av siffran för att isolera det relevanta området och beskära det.

Ändrar storleken på siffran medan aspektkvoten bibehålls och centrerar den i en 28x28 pixelbild.

Normaliserar pixelvärden och plattar ut bilden så att den blir lämplig för modellens inmatning.

7 Slutsatser

1. Hur effektiv är maskininlärning vid igenkänning av handskrivna siffror, och vilka är dess begränsningar?

Maskininlärning är mycket effektiv vid igenkänning av handskrivna siffror, särskilt med avancerade modeller som XGBoost, CNNs och andra djupa inlärningsmetoder. Moderna modeller kan uppnå hög noggrannhet genom att identifiera mönster i handskrivna siffror baserat på pixelvärden och strukturella egenskaper.

Styrkor: Hög noggrannhet de flesta maskininlärningsmodeller kan uppnå över 95 % noggrannhet på dataset som MNIST.
Automatisk funktionsextraktion, särskilt djupa neurala nätverk kan automatiskt identifiera viktiga egenskaper i bilder utan behov av manuell feature engineering.

Begränsningar: Känslighet för brus och otydliga bilder om siffrorna är suddiga eller förvrängda kan modellen ha svårt att göra korrekta förutsägelser.

Datasetsberoende modellen presterar bäst när träningsdata liknar testdata. Om den ställs inför handstilar som avviker kraftigt från träningsdata kan noggrannheten minska.

Sammanfattning: Maskininlärning är mycket effektiv för igenkänning av handskrivna siffror, men dess prestanda beror på datakvalitet, modellval och träningsstrategier. Genom att förbättra datarensning, använda avancerade modeller och optimera hyperparametrar kan begränsningarna minskas.

2. Hur kan den valda modellen förbättras i termer av noggrannhet, effektivitet och generalisering, och vilka alternativa metoder kunde ha använts?

Data Augmentation: Inför små rotationer, förskjutningar eller förvrängningar i träningsdata för att hjälpa modellen att hantera variationer bättre, vilket hjälper till att minska överanpassning och gör modellen mer robust mot verkliga variationer.

Justera hyperparametrar ytterligare: Prova djupare träd (öka max_depth) för bättre igenkänning av komplexa mönster. Finjustera inlärningshastighet och regulariseringsparametrar för att minska bias samtidigt som variansen hålls under kontroll.

Experimentera med Ensemble Learning: Kombinera XGBoost med CNN:s (Convolutional Neural Networks) för djupare funktionsutvinning. Använd stacking-modeller, där XGBoost-förutsägelser matas in i en annan modell som en SVM eller ett djupt neuralt nätverk.

Generalisering: Användning av neurala nätverk: Djupa lärandemodeller, såsom CNN (Convolutional Neural Networks), skulle troligen ha gett ännu bättre resultat eftersom de är specialiserade på bildigenkänning.

Testning på andra dataset: För att säkerställa att modellen fungerar bra utanför MNIST kunde vi ha testat den på andra handskrivna sifferdataset, såsom EMNIST eller SVHN.

8 Teoretiska frågor

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

Träning set är den uppsättning data som används för att träna modellen och den måste inkludera alla möjliga inmatningar som modellen kan bearbeta. Till exempel, om din modell ska klassificera bilder på katter och hundar, måste träningsuppsättningen inkludera både katter och hundar.

Valideringsuppsättningen, ibland kallad utvecklingsuppsättningen, är en mellanliggande uppsättning mellan tränings- och testuppsättningarna. Dess huvudsakliga syfte är att finjustera modellens hyperparametrar och bedöma dess prestanda under träning.

Testuppsättningen är en opartisk referens för att utvärdera modellens prestanda efter träning. Den simulerar verkliga data som modellen troligtvis kommer att stöta på i produktion.

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 ”validerings-dataset”?

Vad hon kan göra är att jämföra prestandan hos de tre modellerna på testuppsättningen och det kan hon göra genom att använda relevanta mått för varje modell

Och om hon vill ha en mer pålitlig utvärdering kan hon tillämpa korsvalidering på träningsdatan innan hon testar på testuppsättningen. Detta ger en mer robust uppskattning av modellernas prestanda.

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

Regression in machine learning refers to a supervised learning technique where the goal is to predict a continuous numerical value based on one or more independent features.

Linear regression: this assumes that there is a linear relationship between the independent and dependent variables. For example, predicting the price of a house based on its size.

Polynomial Regression
It is used to model non-linear relationships. For example, if we want to predict the population growth over time.

Ridge & Lasso Regression
Ridge & lasso regression are regularized versions of linear regression that help avoid overfitting by penalizing large coefficients.

Support Vector Regression (SVR)
SVR works by finding a hyperplane that minimizes the sum of the squared residuals between the predicted and actual values.

Decision Tree Regression

Decision tree regression uses a tree-like structure to make decisions, where each branch of the tree represents a decision, and the leaves represent outcomes. For example, predicting customer behavior based on features like age, income, etc.

Random Forest Regression

Random Forest is an ensemble method that builds multiple decision trees, each trained on a different subset of the training data. The final prediction is made by averaging the predictions of all the trees. For example, customer churn.

4.Hur kan du tolka RMSE och vad används det till:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

RMSE (root mean squared error) kan tolkas som den genomsnittliga skillnaden i +/- som förväntas mellan ett förutsagt värde och det faktiska värdet. Det är standardavvikelsen för residualer (skillnaden mellan det observerade värdet och det förutsagda värdet för en funktion). RMSE mäts i samma enhet som målvärdet.

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

klassificering ett problem med prediktiv modellering där klassetiketten förväntas för ett specifikt exempel på indata. Till exempel, vid bestämning av handskriftstecken, identifiering av skräppost och så vidare, kräver klassificeringen träningsdata med ett stort antal datauppsättningar av indata och utdata. De vanligaste klassificeringsalgoritmerna är binär klassificering, multi-klassklassificering, multi-label-klassificering och obalanserad klassificering till exempel : Logistic Regression, Decision Tree ,Random Forest ,Support Vector Machine (SVM) ,XGBoost.

Confusion Matrix : confusion matrix, also known as error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning

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

K-means är en av de enklaste och mest kända oövervakade inlärningsalgoritmerna . Du kan använda algoritmen för en mängd olika maskininlärningsuppgifter, till exempel: Identifiera onormala data ,Klustring av textdokument , Analysera datamängder innan du använder andra klassificerings- eller regressionsmetoder.

7.Förklara (gärna med ett exempel): Ordinal encoding, one-hot encoding, dummy variable encoding.

Ordinal encoding is a preprocessing technique used for converting categorical data into numeric values that preserve their inherent ordering.

One Hot Encoding is a method for converting categorical variables into a binary format. It creates new columns for each category where 1 means the category is present and 0 means it is not. The primary purpose of One Hot Encoding is to ensure that categorical data can be effectively used in machine learning models.

dummy variable encoding Liknar One-Hot Encoding, men släpper en av kategorierna för att undvika onödigt beroende mellan kolumner

8.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?

Både Göran och Julia har rätt, men på olika sätt, first göran har rätt för att de två huvudsakliga typerna åv kategoriska data är nominal och ordinal. Julia har också rätt för att det bror på hur man tolkar det liksom i exemplet röd,grön , blå är bara etiketer då är det nominal men när man säger röda skorta är lika med vackrast på festen då gav vi datan en ordning eller rangording och den blir ordinal.

9 Självutvärdering

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

Under projektets gång stötte jag på flera utmaningar,

En av de största utmaningarna var att skriva rapporten på svenska, eftersom det var första gången.

När det gäller maskininlärning var den största utmaningen bildförbehandlingen. Trots att jag uppnådde bra resultat på testuppsättningen med den valda och finjusterade modellen, var processen för bildförbehandlings iterativ och krävde många justeringar.

Jag övervägde även att införa en CSV-fil i appen där användarna skulle kunna rätta till felaktiga förutsägelser. Denna CSV-fil skulle sedan kunna användas för framtida träning av modellen. Detta skulle dock ha krävt en betydande mängd användardata för att generera tillräckligt med träningsinformation, och jag kunde inte genomföra denna funktion på grund av tidsbegränsningar. Jag övervägde också att använda dataaugmentation eller manipulera datasetet för att förbättra modellens prestanda, men återigen hindrade tidsbrist mig från att utföra dessa alternativ.

Trots dessa utmaningar gjorde jag framsteg och lärde mig värdefulla läxor under processen. I framtiden, med mer tid, planerar jag att finslipa modellen ytterligare genom att använda dessa tekniker.

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

Jag anser att jag bör få betyget VG eftersom jag har genomfört projektet med noggrant arbete och har lärt mig och tillämpat nya tekniker, trots de utmaningar som uppstod.

3.Något du vill lyfta fram till Antonio?

Jag skulle vilja tacka dig för allt i den här kursen, för allt du har lärt mig och för ditt stöd. Jag skulle vara mycket tacksam om du kan ge mig någon form av feedback på rapporten och koden, så att jag kan se vad jag kan förbättra, eftersom vi kommer att använda samma tillvägagångssätt i nästa kurser. Jag skulle verkligen vilja veta var jag kan göra saker bättre. Tack så mycket

10 Appendix A

Classification Report of Tuned XGBoost Model:

Class	Precission	Recall	F1-Score	Support
0	0,99	0,98	0,99	1343
1	0,98	0,98	0,98	1600
2	0,96	0,97	0,97	1380
3	0,98	0,96	0,97	1433
4	0,96	0,98	0,97	1295
5	0,98	0,98	0,98	1273
6	0,98	0,98	0,98	1396
7	0,97	0,98	0,97	1503
8	0,96	0,96	0,96	1357
9	0,96	0,95	0,96	1420
Accuracy	0,97			14000

Model Architectures and Hyperparameters

XGBoost Classifier (Selected Model)

- subsample: 0.8
- Number of estimators: 100
- Max depth: 10
- learning_rate: 0.1
- colsample_bytree: 0.7

11 Källförteckning

Géron. (2019). i *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow (2nd ed.).* (s. 91).

IBM. (u.d.). Hämtat från https://www.ibm.com/think/topics/decision-trees

IBM. (u.d.). Hämtat från https://www.ibm.com/think/topics/classification-machine-learning

IBM. (u.d.). Hämtat från https://www.ibm.com/think/topics/xgboost