

Bank Customer Churn



Kawser Ayoub

EC Utbildning

Projektkurs

202411

Abstract

This report investigates predicting customer churn in the banking sector using machine learning models with Artificial Neural Networks (ANN) identified as the best performing approach. ANN achieved balanced precision and recall of 78%, highlighting key churn predictors such as the number of products, age and tenure. The study emphasizes integrating AI insights with practical strategies, like analyzing customer feedback and enhancing service, for effective churn management.

Innehållsförteckning

Abstract.....	2
1 Inledning.....	4
2 Teori.....	5
2.1 EDA.....	5
2.2 Data förbehandling.....	5
2.3 Obalanserad datahantering.....	6
2.4 Machine Learning.....	6
2.5 Deep Learning.....	7
2.6 Modellutvärdering.....	8
2.7 Feature importance.....	9
2.8 Grid Search CV.....	9
3 Metod.....	10
3.1 Datasetsamling och förbehandling.....	10
3.2 Klass obalans.....	10
3.3 Data Uppdelning.....	10
3.4 Modell implementering.....	10
3.5 Utvärdering.....	11
3.6 Streamlit.....	11
3.7 Agil arbetsmetodik.....	11
4 Resultat och Diskussion.....	12
4.1 Modell Prestanda.....	12
5 Slutsatser.....	18

1 Inledning

Kundavhopp eller churn är en utmaning för många banker idag. Även om många kunder är nöjda, finns det alltid en grupp som överväger att byta bank. Ibland på grund av bättre erbjudande eller förändrade behov. Att förstå vad som får vissa kunder att lämna kan hjälpa banken att agera mer proaktivt och förbättra kundupplevelsen.

Med hjälp av dataanalys och artificiell intelligens kan vi få en bättre förståelse för vilka faktorer som är kopplade till churn. I stället för att gissa vilka kunder som är på väg att lämna kan banken på ett mer grundläggande sätt identifiera potentiella risker och utvärdera om det finns specifika mönster eller egenskaper som återkommer bland dessa kunder. En modell som förutser kundavhopp blir därmed ett användbart verktyg och det kan ses som ett komplement till andra strategier för kundvård och serviceförbättring.

Syftet med denna rapport är att bygga en modell för att förutspå kundavhopp inom banksektorn och analysera de faktorer som påverkar kundernas beslut att stanna eller lämna.

För att uppfylla syftet så kommer följande frågeställningar att besvaras:

1. Vilka faktorer har störst inverkan på kundavhopp och kan vi kvantifiera deras relativa betydelse i modellen?
2. Kan den utvecklade modellen uppnå en precision och recall på minst 80% för churn-klassen?

2 Teori

2.1 EDA

Vad är EDA? EDA står för Exploratory Data Analysis och är en process som används för att undersöka datauppsättningar och hitta underliggande mönster, relationer och avvikelser. Det innebär att sammanfatta viktiga egenskaper hos data och visualisera dem för att formulera hypoteser för vidare analys. EDA hjälper för att ge initiala insikter utan att göra antaganden, vilket ger en tydligare grund för modellering och beslutsfattande. Sammanfattningsvis är syftet med en EDA är att förstå datans underliggande struktur, identifiera mönster, upptäcka avvikelser och formulera hypoteser för vidare analys.

Tekniker som har använts för detta projekt är: univariat analys, bivariat analys och multivariat analys. Univariat analys undersöker enskilda variabler med hjälp av histogram, stapeldiagram och beskrivande statistik för att förstå fördelningen. Bivariat analys utforskar relationer mellan två variabler, särskilt mellan variabler och målvariabler genom att använda till exempel boxplot och stapeldiagram. Multivariat analys används för att analysera korrelationer mellan flera variabler med hjälp av värmekartor för att identifiera multikollinearitet och samband mellan variabler.

2.2 Data förbehandling

Syftet med data förbehandling är att säkerställa datakvaliteten och förbereder funktioner för optimal modell prestanda. Detta inkluderar metoder som att rensa, skala och transformera data för att göra den lämplig för analys och modellering. I detta projekt hanteras saknade värde, funktionsskalning, standardisering och omvandling av kategoriska variabler.

2.2.1 Funktionsskalning

Skalning av variabler förhindrar inkonsekvenser och hjälper modeller att tolka funktioner korrekt. De två vanligaste metoderna är normalisering och standardisering. För normalisering används MinMaxScaler för att skala värden inom ett konsekvent intervall, vanligtvis 0 till 1. Detta bevarar skillnaderna mellan funktioner samtidigt som värdena hålls jämförbara. För standardisering används StandardScaler för att standardisera funktioner så att de har ett medelvärde på 0 och en standardavvikelse på 1. Den här metoden är användbar för modeller som är känsliga för funktionsskalor som SVM eller logistisk regression och fungerar bäst med normalfördelade variabler.

2.2.2 Kategoriska variabler

De flesta algoritmer kräver numeriska indata, så kategoriska variabler måste transformeras. One Hot Encoding är en metod som konverterar varje kategori till binära vektorer, vilket gör att modellen kan tolka varje kategori som oberoende information utan att införa någon hierarki. Varje kategori får sitt eget fält, behåller ursprungliga utmärkelser och förbättrar modellens noggrannhet.

2.3 Obalanserad datahantering

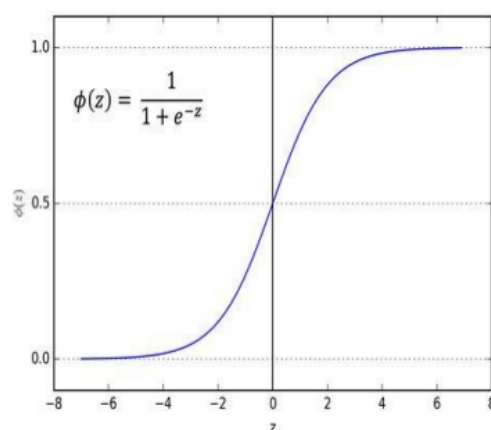
I avhoppss förutsägelser är det vanligt med obalanserade datauppsättningar, där fall som är icke avhopp ofta är långt fler än avhoppss fallen. Denna obalans skapar utmaningar eftersom den kan få modeller att sned ställas mot majoritetsklassen, vilket potentiellt missar mönster i minoritetsklassen (avhoppss kunder). Det krävs att man applicerar en åtgärd som balanserar datan för att skapa en rättvis och korrekt prediktiv modell. Två tekniker applicerar för detta projekt och dessa är:

Synthetic Minority Over-sampling Technique även kallas för SMOTE skapar syntetiska exempel för minoritetsklassen, vilket förbättrar representationen av avhopp fall utan att duplicera befintliga data. Class weighting är ett alternativ till översampling, klassviktning tilldelar minoritets klassen högre vikt inom själva algoritmen (t.ex. genom att sätta `class_weight='balanced'`). Denna metod gör att modellen kan anpassa sitt fokus på avhopp fall naturligt utan att ändra data strukturen.

2.4 Machine Learning

2.4.1 Logistisk Regression

Logistisk regression är en statistisk metod som används för att förutsäga binära utfall, vanligen tillämpad när den beroende variabeln är kategorisk, till exempel ja eller nej. Den uppskattade sannolikheten för att en händelse inträffar genom att använda en logistisk funktion, som producerar värden mellan 0 och 1. (IBM)

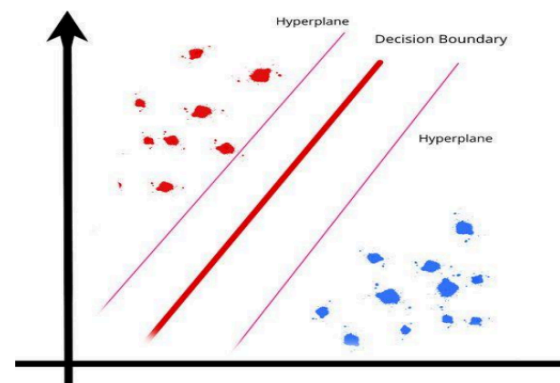


Figur 1. Logistisk Regression algorithm

2.4.2 Support Vector Machine (SVM)

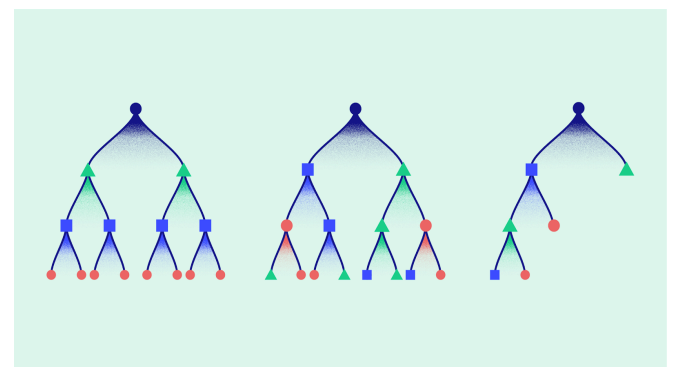
Support Vector Machines används främst för klassificeringsproblem och fungerar genom att hitta ett optimalt hyperplan som maximerar marginalen mellan klasser och separerar datapunkter på ett sätt som minimerar klassificeringsfel. (IBM)

Figur 2. SVM algorithm



2.4.3 Extreme Gradient Boosting (XGB)

XGB är en maskininlärning algoritm inom ensemble learning och känd för sin hastighet och prestanda i klassificerings- och regressions uppgifter. Genom att utnyttja beslutsträd som grundmodeller och inkludera regulariseringstekniker förbättrar XGBoost modellens generalisering och minskar risken för överanpassning.

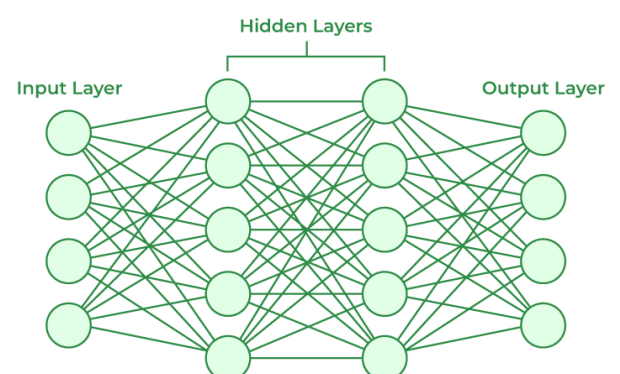


Figur 3. XGB algorithm

2.5 Deep Learning

2.5.1 Artificial Neural Networks (ANNs)

Artificiella neurala nätverk är beräkningsmodeller inspirerade av den mänskliga hjärnan, designade för att känna igen mönster i data. De består av ett input lager, dolda lager och ett output data lager. Varje lager består av "neuroner" som bearbetar information och skickar den till nästa lager, vilket gör det möjligt för nätverket att lära sig komplexa relationer inom data. Det sker med hjälp av att justera vikter och fördomar och gör nätverket mer exakt med tiden. (IBM)



Figur 4. ANNs algorithm

2.5.2 Tränings komponenter

Inom de dolda lagren används aktiverings-funktioner som ReLU, vilket lägger till icke-linjäritet och gör modellen mer flexibel för komplexa mönster. För binär klassificering används ofta sigmoid-funktionen i output lagret för att omvandla resultat till sannolikheter mellan 0 och 1.

För att minska risken för överanpassning används regulariserings-tekniker som dropout, där neuroner slumpmässigt stängs av under träning. Träningen av ett ANN inkluderar en förlustfunktion, oftast binär korentrop för att mäta fel och en optimerare som Adam, som anpassar inlärnings hastigheten för effektiv konvergens. Early stopping kan också användas för att stoppa träningen när validerings-prestandan stabiliseras, vilket leder till en mer generaliserbar modell.

2.6 Modellutvärdering

2.6.1 Accuracy

Accuracy är ett klassificeringsmått som indikerar andelen korrekta förutsägelser av de totala observationerna. Den beräknas som förhållandet mellan korrekta förutsägelser och det totala antalet förutsägelser, vilket ger en översikt över modellens övergripande prestanda. (Iguazio)

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

Figur 5. Accuracy

2.6.2 Precision

Precision är ett klassificeringsmått som mäter andelen sanna positiva förutsägelser ibland alla positiva förutsägelser som modellen gör. Den beräknas genom att dividera antalet sanna positiva med summan av sanna och falska positiva. (Analytics Vidhaya)

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

Figur 6. Precision

2.6.3 Recall (Känslighet)

Recall är ett mått i klassificeringen som mäter förhållandet mellan sanna positiva förutsägelser och de totala faktiska positiva. Den beräknas genom att dividera antalet sanna positiva med summa av sanna positiva och falska negativa, vilket indikerar modellens förmåga att identifiera alla relevanta instanser i datamängden. (Analytics Vidhaya)

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

Figur 7. Recall

2.6.4 F1 Score

F1-score är ett mått som kombinerar precision och recall till en enda mått på noggrannhet. Det beräknas som medelvärde av precision och recall. (Wikipedia)

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

Figur 8. F1 Score

2.6.5 Confusion Matrix

En confusion matrix är en tabell som används i klassificering för att visualisera en modells prestanda genom att visa antalet sanna positiva, sanna negativa, falska positiva och falska negativa. Denna tabell ger insikter om modell noggrannhet i varje klass som hjälper till att identifiera mönster av felklassificering. Varje cell i matrisen motsvarar faktiska förutspådda klass-jämförelser, vilket ger en översikt över modellens prestanda. (IBM)

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figur 9. Confusion Matrix

2.7 Feature importance

2.7.1 Permutation Importance

Permutation Importance används främst inom maskininlärning för att bedöma variabel signifikans. Det fungerar genom att slumpmässigt blanda värden för en variabel och mäta den resulterande effekten på modellens prestanda. Detta ger denna metod insikt i varje variabls bidrag till prediktiv noggrannhet. Permutations importance är användbart i olika sammanhang där det är viktigt att förstå variabls inverkan i komplexa system och erbjuder ett tolkbart sätt att rangordna funktioners betydelse baserat på prestanda påverkan. (scikit learn)

2.8 Grid Search CV

Det är en teknik som främst används inom maskininlärning men även tillämpbar i statistisk modellering för att optimera modellparametrar. Den testar systematiskt kombinationer av specificerade hyperparametrar genom att utvärdera modell-prestanda för varje inställning. (scikit learn)

3 Metod

3.1 Datainsamling och förbehandling

Datainsamlingen erhöles från Kaggle med hjälp av paketet `opendatasets`. Datasetet innehåller 10 000 observationer och 12 variabler som representerar kunders demografi, kontoställningar och churn-status. Datan var komplett och saknade inga värden vilket reducerade behovet av djupare datahantering. Variablerna inkluderar både numeriska och kategoriska data.

Efter datainsamling tas `'customer_id'` bort eftersom det inte är relevant. Targeten `'churn'` separeras medan de återstående variablerna bildas till indata (X). Numeriska kolumnerna `'credit_score'`, `'balance'` och `'estimated_salary'` skalas med `MinMaxScaler` för att normalisera deras värden. Kategoriska variabler som `'country'` och `'gender'` omvandlas till numeriska värden.

3.2 Klass obalans

För att hantera klass obalansen i datasetet, används två metoder beroende på modell:

- SMOTE: Detta används med mer komplexa modeller som XGB, ANNs och SVM för att skapa syntetiska datapunkter för minoritetsgruppen.
- `class_weight='balance'`: För enklare modeller som logistisk regression används istället för klasser i modeller. Detta är särskilt användbart för att hantera obalanser utan att lägga till nya datapunkter.

3.3 Data Uppdelning

För maskininlärning modeller delades data upp i 70% träning, 15% validering och 15% test. SMOTE applicerades på träning setet och ökade dess storlek från 7000 rader till 11 094 rader, medan validerings och test-set förblev oförändrade.

```
Original data: 10000 rows
Training data: 7000 rows
Training data (SMOTE): 11094 rows
Validation data: 1500 rows
Test data: 1500 rows
```

Figur 10. Data uppdelning

För djup inlärningsmodellen delades resampled data upp i 80% träning och 20% test.

3.4 Modell implementering

Flera maskininlärning modeller samt en djupinlärning moell implementerades och jämfördes:

- Logistisk regression: användes som baslinje modell, som inkluderar klass viktning för att åtgärda klass obalans.
- Support Vector Machine: konfigurerad med en RBF-kärna för icke linjär klassificering och använder klass viktning för att åtgärda balans.
- XGBoost: En ensemble modell som utnyttjar gradient-förstärkning, optimerad med SMOTE för balanserad träningsdata och hyperparameter-justering.
- Artificiella neurala nätverk (ANN): Byggt med tre dolda lager, en sigmoid utgång och dropout för att förhindra överanpassning. Det användes Adam optimizer, binär kors-entropi förlust och tidigt stopp för träning.

3.5 Utvärdering

Modellerna utvärderas på test setet med hjälp med följande mätvärden:

- Noggrannhet: Andelen korrekt predikterade observationer av de totala observationerna.
- Precision and Recall: Fokuserad specifikt för att bedöma modellens förmåga att identifiera och korrekt klassificera churn-fall.
- F1-Score: Ett balanserat mått som kombinerar precision och återkallelse.
- Confusion matrix: Ger en detaljerad uppdelning av sanna positiva, sanna negativa, falska negativa och falska positiva och gav insikter om felklassificering mönster.

Resultaten sammanställdes i tabeller och visualiserades för jämförelseanalys.

3.6 Streamlit

Ett interaktivt användargränssnitt utvecklades med Streamlit för att göra modellen tillgänglig och användarvänlig. Gränssnittet gjorde det möjligt för användare att mata in kunddata och ta emot förutsägelser i realtid angående churn-sannolikhet.

3.7 Agil arbetsmetodik

3.7.1 Scrum

Projektet följde en agil metodik och det användes Scrum framework för att hantera utvecklingen.

1. Sprintar: Varje sprint varade i en vecka och fokuserade på distinkta uppgifter såsom datainsamling, modellering och utvärdering.
2. Dagliga stand-ups: Korta dagliga möten hölls för att identifiera utmaningar och planera dagens aktiviteter.
3. Demos: I slutet av varje sprint visades framsteg för gruppmedlemmarna för feedback och justeringar.

4 Resultat och Diskussion

4.1 Modellprestanda

Logistisk Regression

Logistisk Regression uppnår en total test noggrannhet på 72% med hög precision (91%) för fall utan churn (Klass 0). Den kämpar dock med att upptäcka churn (Klass 1) och uppnår endast 39% precision, även om recall för churn fall är relativt högre med 72%. Detta indikerar att modellen ofta fel klassificerar icke churn fall som churn, vilket leder till en avvägning i prestanda.

```
LogReg Validation Accuracy: 0.70
LogReg Test Accuracy: 0.72
```

Figur 11. Logistic Regression Validation & Test Accuracy

LogReg Classification Report (Test):				
	precision	recall	f1-score	support
0	0.91	0.72	0.80	1200
1	0.39	0.72	0.51	300
accuracy			0.72	1500
macro avg	0.65	0.72	0.66	1500
weighted avg	0.81	0.72	0.74	1500

Figur 12. Logistic Regression Classification Report

XGBoost

Utan SMOTE: XGBoost uppnår högre noggrannhet (84%) med stark prestanda för att förutsäga fall utan churn (precision 87%, recall 94%). Modellen kämpar dock med churn-detektering och uppnår endast 53% recall, vilket indikerar att många churn fall missas.

Med SMOTE: XGBoost förbättrar sin förmåga att upptäcka churn, vilket ökar recall till 55%. Den totala noggrannheten sjunker dock till 81% och precisionen för churn fall minskar till 53%, vilket återspeglar en avvägning mellan att balansera klasserna och bibehålla övergripande prediktiv prestanda.

```

XGBoost Validation Accuracy (no SMOTE): 0.85
XGBoost Test Accuracy (no SMOTE): 0.84
XGBoost Classification Report (Test, no SMOTE):
      precision    recall  f1-score   support

     0       0.87       0.94       0.90       1200
     1       0.65       0.45       0.53        300

 accuracy
macro avg       0.76       0.70       0.72       1500
weighted avg       0.83       0.84       0.83       1500

```

Figur 13. XGB Classification Report

```

XGBoost Validation Accuracy (with SMOTE): 0.81
XGBoost Test Accuracy (with SMOTE): 0.81
XGBoost Classification Report (Test, with SMOTE):
      precision    recall  f1-score   support

     0       0.89       0.88       0.88       1200
     1       0.53       0.55       0.54        300

 accuracy
macro avg       0.71       0.71       0.71       1500
weighted avg       0.82       0.81       0.81       1500

```

Figur 14. XGB Classification Report (SMOTE)

Support Vector Machine

Utan SMOTE: SVM uppnår en test noggrannhet på 79% och presterar bra när det gäller att upptäcka fall utan churn (precision 92%, recall 81%). Dess prestanda för churn fall är dock begränsad, med ett recall på 74% men låg precision på 49%, vilket indikerar frekventa falska positiva resultat för churn.

Med SMOTE: SVM förbättrar sin förmåga att balansera förutsägelser och uppnår en högre test noggrannhet på 82%. Recall för klass 1 minskar något till 63% men precision förbättras till 55%, vilket återspeglar en mer balanserad modell som minskar falska positiva resultat för churn samtidigt som man effektivt förutsäger fall utan churn.

```

SVM Validation Accuracy: 0.79
SVM Test Accuracy: 0.79
SVM Classification Report:
      precision    recall  f1-score   support

     0       0.92       0.81       0.86       1200
     1       0.49       0.74       0.59        300

 accuracy
macro avg       0.71       0.77       0.72       1500
weighted avg       0.84       0.79       0.81       1500

```

Figur 15. SVM Classification Report

```

SVM Validation Accuracy (SMOTE): 0.82
SVM Test Accuracy (SMOTE): 0.82
SVM Classification Report (Test (SMOTE):
      precision    recall  f1-score   support

      0       0.90      0.87      0.89     1200
      1       0.55      0.63      0.59      300

   accuracy          0.82     1500
  macro avg       0.73      0.75      0.74     1500
 weighted avg       0.83      0.82      0.83     1500

```

Figur 16. SVM Classification Report (SMOTE)

Bästa maskinginlärningsmodellen

Vi har funnit att SVM med SMOTE är den bäst presterande modellen baserat på jämförelsen av noggrannhet, recall och precision.

Efter finjustering förblir modellens prestanda liknande som tidigare. Testnoggrannheten förblev nästan samma, 82%, med en liten minskning i precision för klass 1 (från 55% till 54%) och en liten förbättring av recall (från 63% till 65%). Överlag gjorde finjusteringen minimal inverkan.

```

Test Accuracy: 0.8193333333333334
Classification Report (Test):
      precision    recall  f1-score   support

      0       0.91      0.86      0.88     1200
      1       0.54      0.65      0.59      300

   accuracy          0.82     1500
  macro avg       0.72      0.75      0.74     1500
 weighted avg       0.83      0.82      0.83     1500

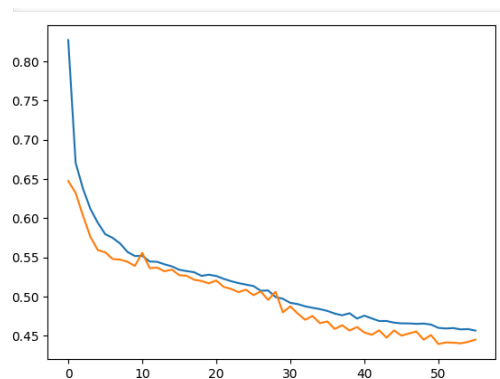
```

Figur 17. Best Model Classification Report

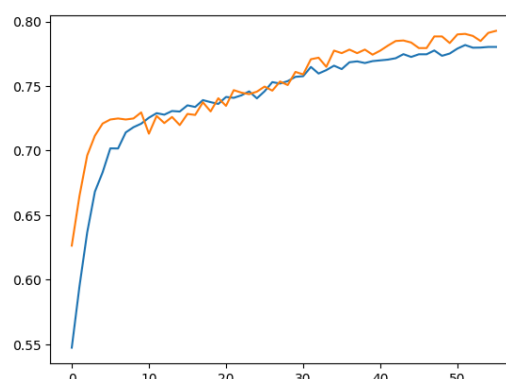
Artificiell neurala nätverk

ANN-modellen uppnådde en total noggrannhet på 78% med en balanserad precision, recall, F1-score över båda klasserna (78%). AUC-score på 78% indikerar vidare en tillförlitlig förmåga att skilja mellan churn och non churn fall. Graferna för förlust och noggrannhet visar konsekvent träning med validering-prestanda som matchar träningen, vilket tyder på minimal överanpassning.

Jämfört med SVM-modellen med SMOTE visar ANN något lägre noggrannhet men ger mer balanserade mätvärden för både precision och recall. Detta tyder på att ANN bättre generaliserar till båda klasserna, vilket gör det till en förbättring när det gäller att fånga klass balansen samtidigt som den bibehåller prediktiva prestanda.



Figur 16. Loss och val loss graf



Figur 17. Accuracy and val accuracy graf

	precision	recall	f1-score	support
0	0.75	0.81	0.78	1534
1	0.81	0.75	0.78	1652
accuracy			0.78	3186
macro avg	0.78	0.78	0.78	3186
weighted avg	0.78	0.78	0.78	3186

AUC: 0.7832464146023468

Figur 18. ANNs Classification Report

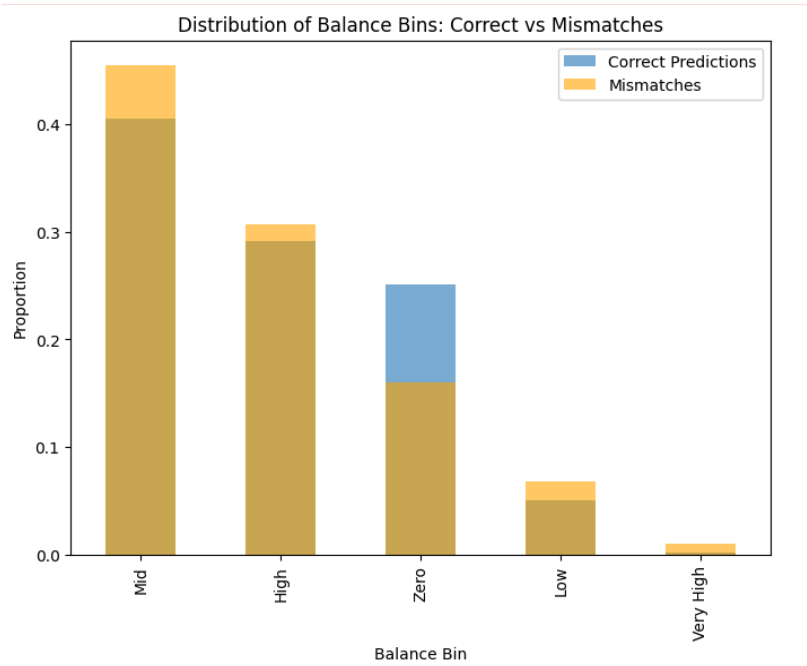
Figur 19. AUC Score

Streamlit implementings-problem och modellanalys

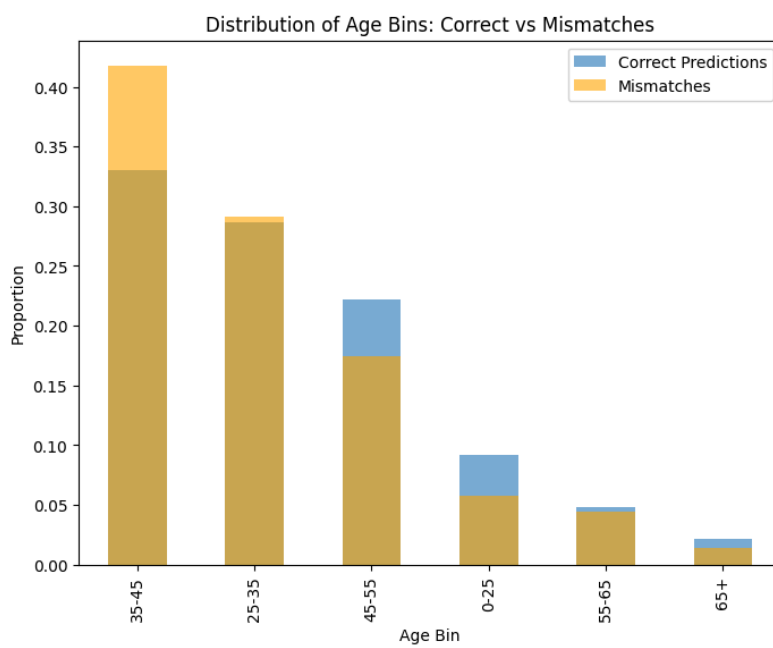
Under implementeringen av djupinlärnings-modellen via Streamlit uppstod ett problem där bra kunder (icke churn) felaktigt flaggades med varningar. För att identifiera orsaken, verifierade vi backend-logiken, inklusive skalning av numeriska variabler och kodningen av kategoriska variabler, genom att implementera loggning. Efter att ha bekräftat att koden och transformationen fungerade korrekt gjorde vi en djupare analys av själva modellen.

Analyssteg:

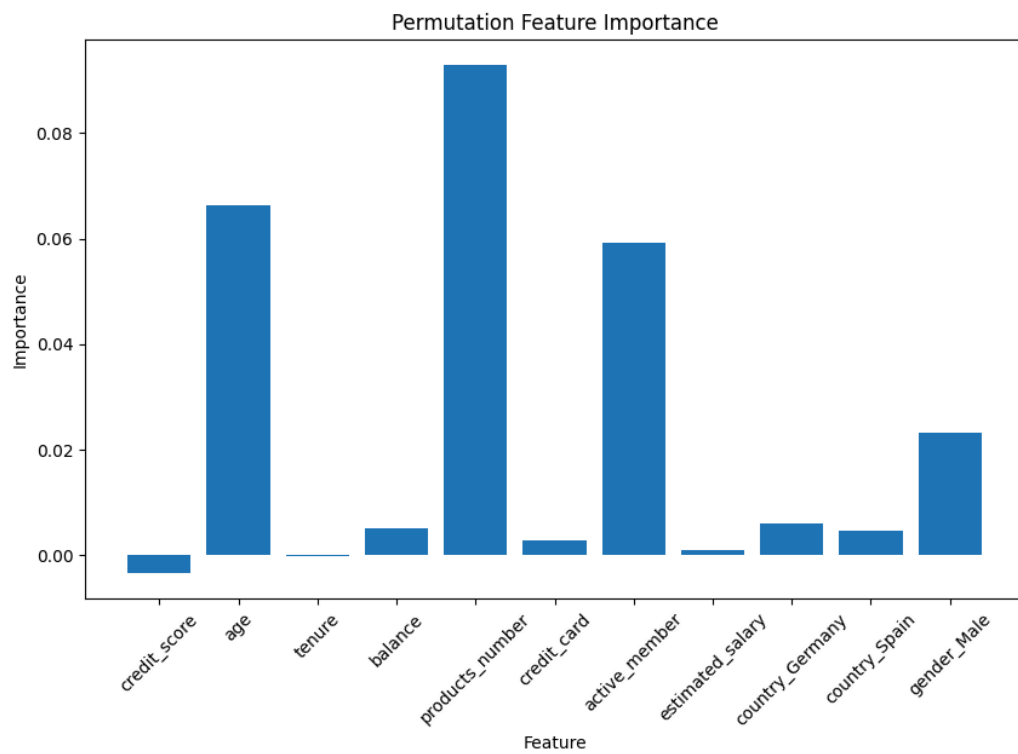
1. Kolmogorov-Smirnov (KS) Test: Signifikanta avvikelser (p-värden $< 0,05$) hittades för balans och ålder, vilket indikerar att modellen har det svårt med dessa variabler. Andra variabler som credit_score och tenure visade inga problem.
2. Balance Analysis: Kunder mellan nollsaldo klassificerades ofta fel, medan mellan klassen hade bättre noggrannhet.
3. Age Analysis: Höga missklassificeringar förekom i 35-45 års ålder behållaren, medan andra åldersgrupper förutspåddes mer exakt.
4. Permutation feature importance: Vikten av denna analys visade att antalet produkter är den mest inflytelserika variabeln i modellens förutsägelser. Ålder och aktiva medlemmar spelar också avgörande roller, vilket framhäver deras betydelse för att bestämma resultat.



Figur 20. Distribution av balance: Rätt vs fel prediktioner



Figur 21. Distribution av age: Rätt vs fel prediktioner



Figur 22. Feature importance stapeldiagram

5 Slutsatser

Den här rapporten syftar till att utveckla en prediktiv modell för att hantera kundavhopp inom banksektorn och analysera de faktorer som påverkar avhopp. Med hjälp av att utforska olika modeller för maskininlärning och djupinlärning, strävade vi efter att identifiera det bästa tillvägagångssättet samtidigt som vi utvärderade de bidragande faktorernas relativa betydelse. Projektet tog upp två nyckelfrågor:

1. Vilka faktorer har störst inverkan på kundavhopp och kan vi kvantifiera deras relativa betydelse i modellen?

Analyser visade att antalet produkter som innehas av kunder var den mest inflytelserika variabeln för att förutsäga churn, följt av ålder och tenure. Dessa insikter, härledda från permutation feature importance, visar det praktiska värdet av att identifiera nyckelfaktor för att behålla kunder.

2. Kan den utvecklade modellen uppnå en precision och recall på minst 80% för churn-klassen?

Bland de testade modellerna visade ANN den mest balanserade prestandan över churn och icke churn klasser. Resultatet indikerar ANN:s förmåga att generalisera effektivt och samtidigt bibehålla rättvisa i klass förutsägelser. Även om den inte uppnådde målet 80%, gör dess konsekventa och balanserade prestanda över alla mätvärden den till den mest lämpliga modellen för denna uppgift. I jämförelse visade SVM med SMOTE något högre total noggrannhet, men led av obalanserade klass prediktioner, särskilt i form av falska positiva resultat och lägre precision för churn klassen. Därför, även om SVM är effektiv för specifika mätvärden, gör ANN balans och generaliserbarhet den till den bästa modellen för denna tillämpning.

Bredare perspektiv

Utöver avancerade AI modeller betonar den här rapporten vikten av att överväga alternativa, icke tekniska lösningar för att hantera kundförlust. Datadrivna förutsägelser är värdefulla men praktiska åtgärder kan komplettera AI insikter för att förbättra kund behållningen:

- **Engagera sig med kundtjänst:** Konversationer med kundtjänst representanter kan avslöja problem som driver churn, vilket ger praktiska insikter.
- **Analysera kundfeedback:** Feedback från undersökningar eller recensioner kan peka på missnöje eller ouppfyllda behov.
- **Undersöka användningsmönster:** Att övervaka hur kunder interagerar med tjänster kan hjälpa till att identifiera områden för att förbättra eller främja variabler som inte används särskilt mycket av kunder.
- **Granska kund falls anteckningar:** Att förstå historiska kundinteraktioner kan avslöja missnöje mönster. Dessutom kan implementering av proaktiva åtgärder, såsom

riktade e-postkampanjer för att utbilda kunder om underutnyttjade verktyg och funktioner, avsevärt förbättra tillfredsställelsen och minska churn.

Slutliga tankar

Även om AI-drivna modeller som ANN ger robusta och balanserade insikter om kundavhopp, bör dessa integreras med praktiska tillvägagångssätt för de mest effektiva resultaten. En kombination av prediktiva analyser och strategier för direkt kundengagemang säkerställer ett holistiskt tillvägagångssätt för att mildra churn och förbättra kund behållningen i banksektorn.

Källförteckning

1. IBM, What is logistic regression?, Tillgänglig på:
<https://www.ibm.com/topics/logistic-regression> (hämtad 31/10-24)
2. IBM, What are support vector machines (SVMs)?, Tillgänglig på:
https://www.ibm.com/topics/support-vector-machine?mhsrc=ibmsearch_a&mhq=support%20vector%20machine (hämtad 31/10-24)
3. Analytics Vidhya, What is XGBoost Algorithm?, Tillgänglig på:
<https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/> (hämtad 31/10-24)
4. IBM, What is a neural network?, Tillgänglig på:
<https://www.ibm.com/topics/neural-networks> (hämtad 31/10-24)
5. Iguazio, What is Model Accuracy in Machine Learning?, Tillgänglig på:
<https://www.iguazio.com/glossary/model-accuracy-in-ml/> (hämtad 31/10-24)
6. Analytics Vidhya, Precision and Recall in Machine Learning, Tillgänglig på:
<https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning/> (hämtad 31/10-24)
7. Wikipedia, F-Score, Tillgänglig på: <https://en.wikipedia.org/wiki/F-score> (hämtad 31/10-24)
8. IBM, What is a confusion matrix?, Tillgänglig på:
<https://www.ibm.com/topics/confusion-matrix> (hämtad 31/10-24)
9. scikit learn, 4.2. Permutation feature importance, Tillgänglig på:
https://scikit-learn.org/1.5/modules/permutation_importance.html (hämtad 31/10-24)
10. scikit learn, GridSearchCV, Tillgänglig på:
https://scikit-learn.org/dev/modules/generated/sklearn.model_selection.GridSearchCV.html (hämtad 31/10-24)