Zorg-Los EDA Rapport

Integratie Opdracht

Sjoerd van Dorp

1032109

BIMCVD10R3

Inhoud

[Opdracht 1: DATA ANALYSE 3](#_Toc155990497)

[Technisch 3](#_Toc155990498)

[Informatie behoefte business 5](#_Toc155990499)

[Opdracht 2: MODEL BUILDING 8](#_Toc155990500)

[Technisch 8](#_Toc155990501)

[Informatie behoefte business 9](#_Toc155990502)

[Opdracht 3: ORGANISATIE EN PRODUCTIE AANBEVELING 14](#_Toc155990503)

[Technisch 14](#_Toc155990504)

[Informatie behoefte business 15](#_Toc155990505)

# Opdracht 1: DATA ANALYSE

## Technisch

1. **Geef aan hoe je de EDA hebt uitgevoerd en met welke tools.**

De EDA is uitgevoerd op een dataset om een inzicht te krijgen in de gegevens. Mijn aanpak was het gebruik van verschillende tools en libraries in Python. Deze libraries zijn:

* numpy voor numerieke bewerkingen
* pandas voor gegevensmanipulatie en -analyse
* seaborn voor datavisualisatie
* matplotlib.pyplot voor het maken van plots
* scipy.stats voor statistische functies

Als eerste begon ik met het inlezen van de datasets. Hier worden de trainings- en testdatasets ingelezen met behulp van de pandas-library:

# Import van datasets  
traindf = pd.read\_csv('1/train.csv')  
testdf = pd.read\_csv('1/test.csv')

Hierna begon ik met het printen van informatie over de dataset. Deze code geeft informatie over de dataset, zoals het type van elke kolom, aantal niet-nul waarden en aantal unieke waarden:

# Print dataset informatie

print("\nDataset informatie:")  
print(traindf.info())  
print("\nStatistische waarden:")  
print(traindf.describe())  
print("\nOntbrekende waarden in Data Set:")  
print(traindf.isnull().sum())

Na het inlezen van de dataset werd het tijd om de data te prepareren. Als eerste begon ik met het droppen van de onnodige kolommen en de “NA” waarden uit verschillende kolommen te verwijderen:

# Drop onnodige kolommen  
print("\nOnnodige kolommen droppen...")  
columns\_to\_drop = ['case\_id', 'patientid']  
traindf = traindf.drop(columns=columns\_to\_drop)  
testdf = testdf.drop(columns=columns\_to\_drop)  
  
# NA waarden in de Bed Grade kolom voor de Train en Test datasets vervangen met de modus  
print("\nLege waarden in 'Bed Grade' en 'City\_Code\_Patient' verwijderen...")  
traindf['Bed Grade'].fillna(traindf['Bed Grade'].mode()[0], inplace = True)  
testdf['Bed Grade'].fillna(testdf['Bed Grade'].mode()[0], inplace = True)  
  
# NA waarden in de City\_Code\_Patient kolom voor de Train en Test datasets vervangen met de modus  
traindf['City\_Code\_Patient'].fillna(traindf['City\_Code\_Patient'].mode()[0], inplace = True)  
testdf['City\_Code\_Patient'].fillna(testdf['City\_Code\_Patient'].mode()[0], inplace = True)

Op het einde paste ik labelcodering toe op de Stay kolom in de trainingsdataset, voegde ik een dummy kolom toe in de testdataset, labelcoderde ik alle kolommen in beide datasets en printe ik alle ontbrekenden warden in de dataset om te laten zien dat de data goed is transformeert;

# Labelcodering van de kolom 'Stay' in de trainingsdataset  
print("\nTekst naar numerieke waarden omzetten in train dataset...")  
le = LabelEncoder()  
traindf['Stay'] = le.fit\_transform(traindf['Stay'].astype('str'))  
  
# Het invoegen van een dummy 'Stay'-kolom in de testdataset om samen te voegen met de trainingsdataset.  
print("\nDummy waarden in Stay kolom zetten...")  
testdf['Stay'] = -1  
df = pd.concat([traindf, testdf])  
df.shape  
  
# Labelcodering toepassen op alle kolommen in de trainings- en testdatasets.  
print("\nTekst naar numerieke waarden omzetten in train en test dataset...")  
for i in ['Hospital\_type\_code', 'Hospital\_region\_code', 'Department',  
 'Ward\_Type', 'Ward\_Facility\_Code', 'Type of Admission', 'Severity of Illness', 'Age']:  
 le = LabelEncoder()  
 df[i] = le.fit\_transform(df[i].astype(str))  
  
print("\nOntbrekende waarden in Data Set:")  
print(traindf.isnull().sum())

1. **Geef aan hoe je structureel transformatie en EDA kan uitvoeren (dus niet eenmalige handmatige handelingen).**

Door de transformatie van de data in mijn EDA, kunnen er nieuwe datasets met dezelfde kolommen worden verwerkt zonder dat er handmatig aanpassingen gemaakt moeten worden

## Informatie behoefte business

1. **Wat zijn de verschillende EDA uitkomsten op gebruikelijke categorieën (bv. lege velden etc.)? Doe hiervoor onderzoek want er zijn er een aantal.**

De uitkomsten van bijvoorbeeld lege velden en waarden worden gevonden in bijlage van data transformatie

# Print dataset informatie

print("\nDataset informatie:")  
print(traindf.info())  
print("\nStatistische waarden:")  
print(traindf.describe())  
print("\nOntbrekende waarden in Data Set:")  
print(traindf.isnull().sum())

Na het runnen van deze code ziet de dataset er zo uit:  
Ontbrekende waarden in Data Set:

*case\_id 0*

*Hospital\_code 0*

*Hospital\_type\_code 0*

*City\_Code\_Hospital 0*

*Hospital\_region\_code 0*

*Available Extra Rooms in Hospital 0*

*Department 0*

*Ward\_Type 0*

*Ward\_Facility\_Code 0*

*Bed Grade 113*

*patientid 0*

*City\_Code\_Patient 4532*

*Type of Admission 0*

*Severity of Illness 0*

*Visitors with Patient 0*

*Age 0*

*Admission\_Deposit 0*

*Stay 0*

*dtype: int64*

Na het runnen van deze code in bijlage 3 om de onnodige kolommen te droppen en NA waarden te vullen, ziet de dataset er zo uit:

Ontbrekende waarden in Data Set:

*Hospital\_code 0*

*Hospital\_type\_code 0*

*City\_Code\_Hospital 0*

*Hospital\_region\_code 0*

*Available Extra Rooms in Hospital 0*

*Department 0*

*Ward\_Type 0*

*Ward\_Facility\_Code 0*

*Bed Grade 0*

*City\_Code\_Patient 0*

*Type of Admission 0*

*Severity of Illness 0*

*Visitors with Patient 0*

*Age 0*

*Admission\_Deposit 0*

*Stay 0*

*dtype: int64*

1. **Beschrijf het data cleaning en preparation proces indien van toepassing.**

Ik begon met het importen van de train dataset en deze op te splitsen tot een train en test set. Door dit bestand op te splitsen kon ik beter resultaat krijgen op de uitgevoerde EDA.

Verder heb ik van de stay, age, Hospital\_type\_code, Hospital\_region\_code, Department, ward\_type, Ward\_Facility\_Code, Type of Admission en Severity of Illness de waarden omgezet naar numerieke waarden om beter een analyse te kunnen maken.

1. **Beschrijf en visualiseer de conclusie en aanbeveling op business, technisch en data niveau.**

Op de heatmap, zie bijlage 5: Heatmap, is te zien dat er op plekken zoals Ward\_Facility\_Code-Hospital\_region\_code het nog niet helemaal goed gaat maar op plekken zoals Visitors With Patient-Stay wel.

In de boxplot is te zien dat de waarden van de vrije kamers en bezoekers per patient dicht bij elkaar liggen, maar er wel een aantal flinke uitschieters te zien zijn waar rekening mee gehouden moet worden. Zie: Bijlage 6: Boxplot.

# Opdracht 2: MODEL BUILDING

## Technisch

1. **Geef aan hoe je de feature engineering hebt gedaan.**

In mijn EDA is er feature engineering gedaan op de volgende manier:

* Kolommen droppen
* NA Values opvullen
* Label Encoding
* Outlier Detectie and Verwijderen

Zie: bijlagen 3 & 7

1. **Licht toe welke tools je onderzocht of gebruikt hebt voor het maken van een model.**

De tools die ik gebruikt heb zijn de volgende:

* Scikit-learn
  + De Scikit-learn library is gebruikt bij het maken van de Naive Bayes en de Gradient Boosting Model.
* XGBoost
  + Het XGBoost-model is gebruikt.voor het gradient boosting model.
* Sklearn
  + Met gebruikt van Sklearn heb ik mijn RandomForest en Naive Bayes model kunnen maken

1. **Leg uit hoe je verschillende visualisaties hebt gemaakt.**

In mijn EDA heb ik 5 plot visualisaties gemaakt. Deze visualisaties zijn:

* Normale Verdeling
  + Een histogram voor de normale verdeling wordt gemaakt voor de kolom 'Admission\_Deposit'. De histplot-functie van seaborn wordt gebruikt om het histogram te maken, en vervolgens wordt de normale verdeling getoond met behulp van norm.pdf van scipy.stats. Zie bijlage 10: *Normaal Verdeling*
* Heatmap
  + Een heatmap wordt gemaakt om de correlatie tussen verschillende kenmerken te visualiseren. De heatmap-functie van seaborn wordt gebruikt.
* Boxplot
  + Een boxplot wordt gemaakt voor geselecteerde kolommen 'Available Extra Rooms in Hospital' en 'Visitors with Patient'. De boxplot-functie van seaborn wordt gebruikt. Zie bijlage 6: *Boxplot*
* AOC Curve
  + AOC-curves worden gemaakt voor elk van de klassen van het XGBoost-model. De roc\_curve-functie wordt gebruikt voor het berekenen van de ROC-curve, en de resultaten worden geplot met matplotlib. Zie bijlage 8: *AOC Curve*
* Confusion Matrix
  + Een confusion matrix wordt gemaakt voor het XGBoost-model om de prestaties van het model te evalueren. De confusion\_matrix-functie wordt gebruikt, en de resultaten worden geplot met seaborn. Zie bijlage 9: *Confusion Matrix*

## Informatie behoefte business

1. **Beschrijf de uitkomsten en conclusie van de feature engineering.**

In de code is dit te zien met de ‘print’ functie tussen de functies door. Dit is hoe het er uit komt te zien: zie bijlage 11: *Uitkomsten Feature Engineering*

1. **Maak een feature heatmap (visualisatie).**

Een heatmap wordt gemaakt om de correlatie tussen verschillende kenmerken te visualiseren. De heatmap-functie van seaborn wordt gebruikt. Zie bijlage 5: *Heatmap*

1. **Welke tool(s) heb je gebruik om een model te testen en wat zijn de conclusies.**

De tools die ik gebruikt heb zijn de volgende:

* Scikit-learn
  + De Scikit-learn library is gebruikt bij het maken van de Naive Bayes en de Gradient Boosting Model.
* XGBoost
  + Het XGBoost-model is gebruikt voor het gradient boosting model.
* Sklearn
  + Met gebruikt van Sklearn heb ik mijn RandomForest en Naive Bayes model kunnen maken

In de conclusie is te zien dat XGboost het beste scoort. Zie bijlage 12: *Model Scores*

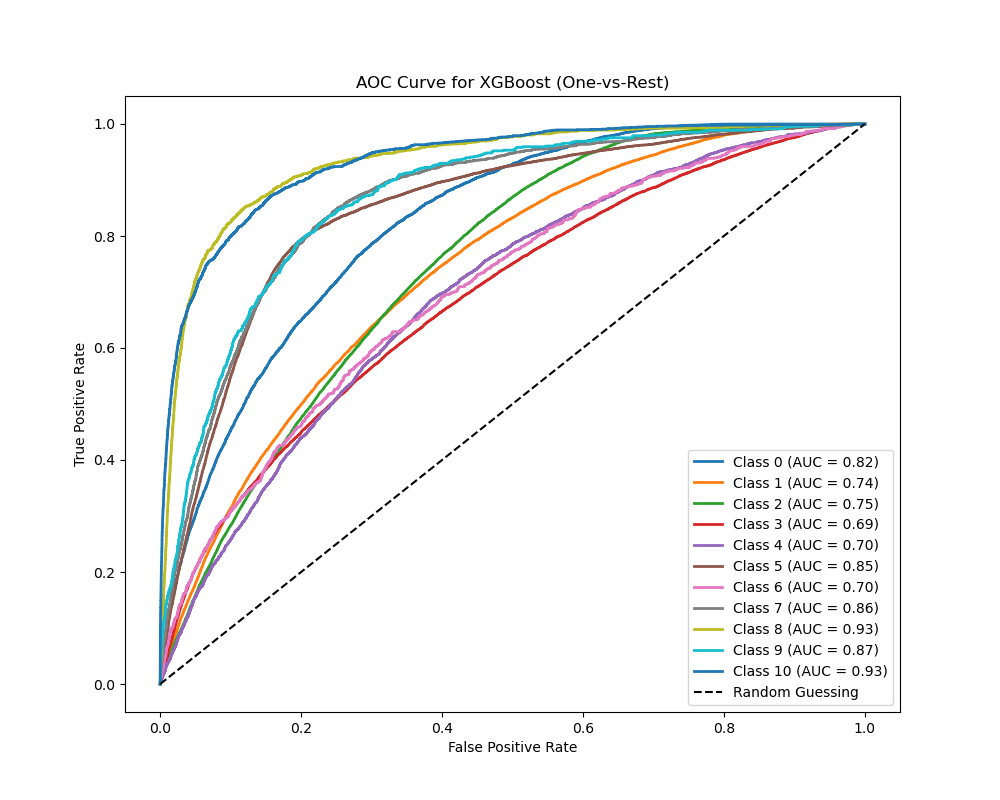
1. **Wat zijn de resultaten van de verschillende modellen en wat is de beste (resultaat in tabel vorm)?**

De tools die ik gebruikt heb zijn de volgende:

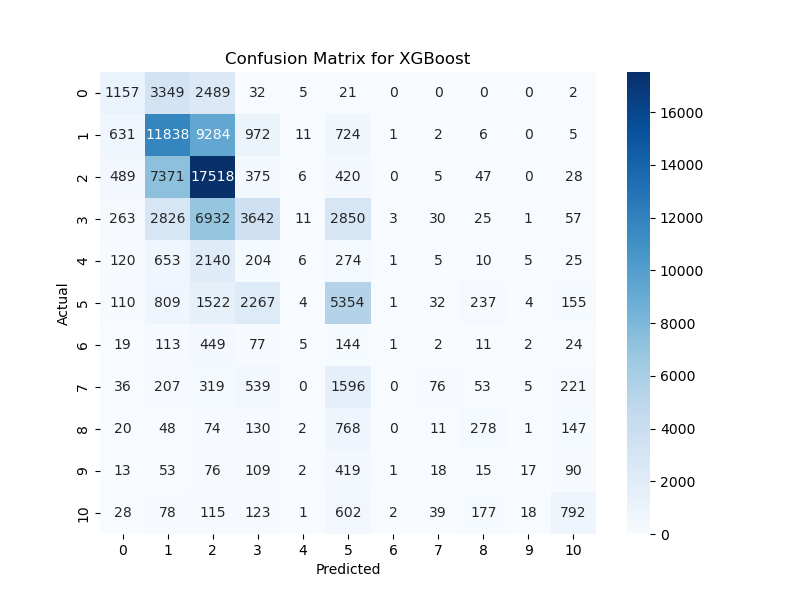
* Scikit-learn
  + De Scikit-learn library is gebruikt bij het maken van de Naive Bayes en de Gradient Boosting Model.
* XGBoost
  + Het XGBoost-model is gebruikt.voor het gradient boosting model.
* Sklearn
  + Met gebruikt van Sklearn heb ik mijn RandomForest en Naive Bayes model kunnen maken

In de conclusie is te zien dat XGboost het beste scoort. Zie bijlage 12: *Model Scores*

1. **Een goede validatie en een confusion matrix, AUC, nauwkeurigheidspercentage en eventueel andere statistisch informatie of conclusies.**

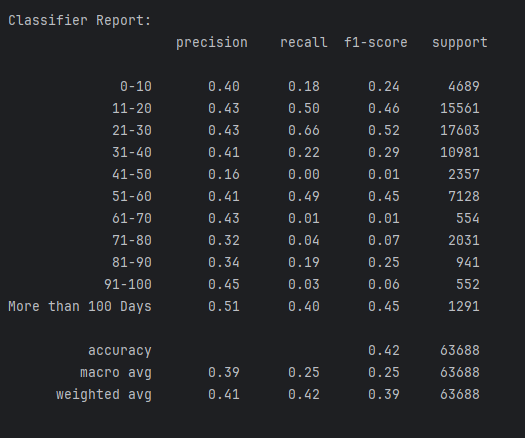
De AOC curve van mijn analyse is hier te zien:  


Een AOC van 1.0 beschrijft een perfect model, terwijl een AOC van 0.5 niet beter is dan willekeurig gokken (zoals aangegeven door de gestippelde lijn). De AOC waarden voor de verschillende klassen variëren, met de hoogste AOC voor Klasse 10 wat wijst op een zeer goede prestatie voor die klasse, en de laagste AOC voor Klasse, wat suggereert dat het model minder nauwkeurig is in het identificeren van die specifieke klasse.

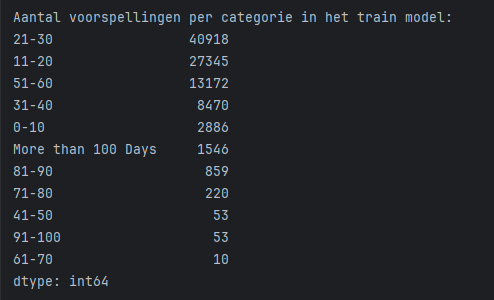
Hier is mijn confusion matrix:  


Voor klasse 1 voorspelde het model 11,838 keer correct, maar het verwisselde klasse 1 ook 631 keer met klasse 0. De kleurgradaties helpen om snel te identificeren welke klassen het beste en slechtste presteren. Zo'n matrix als dit is heel nuttig om te bepalen welke klassen mogelijk meer aandacht nodig hebben tijdens het trainen van het model.

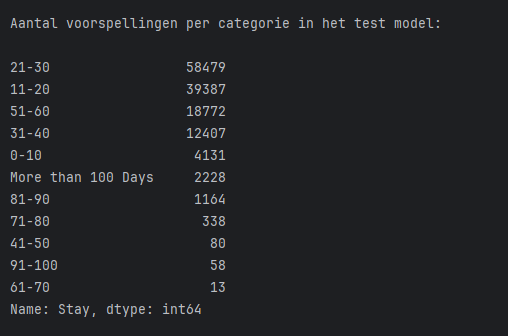
De nauwkeurigheidspercentage van mijn model is hier te zien:



Precision meet het aandeel correcte positieve voorspellingen ten opzichte van alle positieve voorspellingen die het model doet. Accuracy daarentegen is het percentage van alle voorspellingen dat het model correct maakt. Het classificatierapport toont aan dat het model in sommige gevallen redelijk precies is, maar over het algemeen heeft het een accuracy van 42%, wat aangeeft dat minder dan de helft van alle voorspellingen correct zijn. Dit suggereert dat er ruimte voor verbetering is in de prestatie van het model.

Het XGboost model heb ik verder uitgewerkt met de test dataset om te kijken wat de voorspellingen zijn. Zo zijn dit de resultaten van het model met gebruik van de train set:  


En dit voor het test set:



Deze voorspellingen laten zien wat het aantal voorspellingen zijn uit beide datasets. Zoals je kan zien zijn de categorische waarden terug omgezet naar de originele versies voor de transformatie.

# Opdracht 3: ORGANISATIE EN PRODUCTIE AANBEVELING

## Technisch

1. **Zorg dat je duidelijk beschrijf hoe de tools zijn toegepast en dat het geheel reproduceerbaar is voor een andere medewerker**

Gebruikte Tools en Technieken:

* Machine Learning Modellen:
  + Gebruik van sklearn modellen zoals GaussianNB, RandomForestClassifier, AdaBoostClassifier, GradientBoostingClassifier, LogisticRegression, KNeighborsClassifier, DecisionTreeClassifier en SVC.
  + Toepassing van GridSearchCV voor hyperparameter tuning, cross\_val\_score en StratifiedKFold voor modelvalidatie.
* Data verwerken:
  + Inzet van QuantileTransformer voor het transformeren van features.
  + Gebruik van LabelEncoder voor het coderen van categorische variabelen.
  + Het terug zetten van de orginele categorien bij de ‘Stay’ waarde om het beter af te kunnen lezen.
* Model Evaluatie:
  + Implementatie van roc\_curve, auc, confusion\_matrix, accuracy\_score en precision\_score voor het evalueren van modelprestaties.

Reproduceerbaarheid:

* Gedetailleerde Code Documentatie:
  + Elk deel van de code, inclusief data preprocessing, modeltraining en evaluatie, is voorzien van commentaar om de functionaliteit en het doel ervan uit te leggen.
* Modulaire Code Structuur:
  + De code is opgedeeld in functies en modules voor specifieke taken (zoals data laden, preprocessing, modeltraining, etc.), wat het hergebruik en aanpassing vereenvoudigt.
* Trainingsmaterialen:
  + Ontwikkeling van documentatie of handleidingen die de logica achter de modelkeuzes en de stappen van de data-analyse uitleggen.

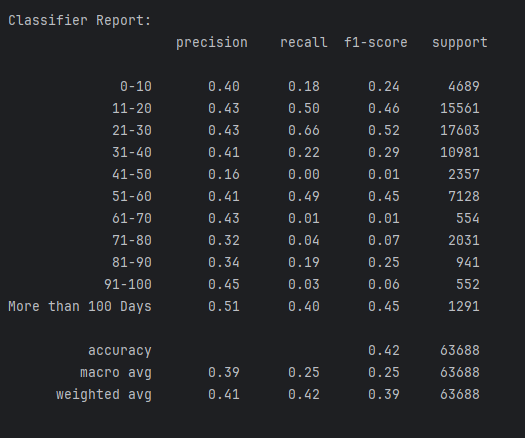
## Informatie behoefte business

1. **Welke tool(s), producten en applicatie/infrastructuur landschap er nodig om dit in te richten?**

* Python Libraries:
  + numpy: Gebruikt voor lineaire algebra en numerieke berekeningen.
  + pandas: Essentieel voor data processing en het lezen van CSV-bestanden.
  + seaborn en matplotlib.pyplot: Voor data visualisatie en het plotten van grafieken.
  + datetime en collections: Voor het werken met datums en datastructuren.
  + sklearn (Scikit-learn): Voor preprocessing van data, modelselectie, training en validatie.
  + classification\_report: De classification\_report om het model te gebruiken om tot de gewenste uitkomst te komen.
* Data Analyse en Verwerking:
  + Python en Bijbehorende Bibliotheken: zoals hierboven genoemd, voor data analyse, verwerking en visualisatie.
* Machine Learning Ontwikkeling:
  + Scikit-learn: Voor het ontwikkelen van machine learning modellen, inclusief preprocessing, model training en evaluatie.
* Ontwikkelomgeving:
  + Een omgeving zoals PyCharm, Jupyter Notebook of Visual Studio Code voor het schrijven en testen van Python code.
* Data Opslag:
  + Gezien het gebruik van pandas voor het lezen van CSV-bestanden, is opslag nodig voor deze datasets, hetzij lokaal of in de cloud (bijv. AWS S3, Azure Blob Storage).

1. **Welke adviezen zijn er te geven op organisatieniveau (onderbouw je aanbevelingen en gebruik literatuur als de intelligente organisatie) over de vervolgstappen?**

Gebaseerd om op de bevindingen van de EDA en de Classifier Report kunnen de volgende aanbevelingen gemaakt worden.



*bijlage 14: Classifier Report*

* Datagedreven besluitvorming: De resultaten tonen het belang van datagestuurd besluitvorming. De organisatie moet gebruikmaken van de beschikbare gegevens om strategische beslissingen te nemen. Dit is in lijn met de principes van 'De Intelligente Organisatie' waarbij beslissingen worden ondersteund door data-analyse en modellering.
* Verbetering van datakwaliteit: De aanwezigheid van ontbrekende waarden en de noodzaak van data schoonmaken benadrukt het belang van het onderhouden van hoge datakwaliteit. De organisatie moet systemen en processen implementeren om de nauwkeurigheid en volledigheid van de data te garanderen (Redman, "Data's Credibility Problem," 2013).
* Focus op belangrijke voorspellers: De analyse onthult welke variabelen het meest voorspellend zijn voor uitkomsten. Dit kan leiden tot gerichte verbeteringen in bepaalde gebieden, zoals het beheer van bepaalde afdelingen of het optimaliseren van patiëntstromen.
* Toepassen van predictive analytics: De resultaten van modellen van XGBoost kunnen worden gebruikt voor predictieve analyses. Dit kan helpen bij capaciteitsplanning en efficiëntieverbetering.
* Training en cultuurverandering: Om een datagedreven benadering te implementeren, is het essentieel dat medewerkers getraind worden in het gebruik en interpretatie van data en analytische tools. Het creëren van een cultuur waarin data wordt gewaardeerd als een belangrijke asset is cruciaal (Davenport, 2006).

1. **Welke AI tools zouden het proces verder en slimmer kunnen automatiseren?**

Het is lastig om specifiek te zeggen welke tools er nodig zijn, dit komt omdat er tegenwoordig al heel veel soorten AI tools zijn. Hier zijn een aantal mogelijke tools die gebruikt kunnen worden:

* Natural Language Processing (NLP): Tools voor het verwerken van taal kunnen worden gebruikt voor het analyseren van ongestructureerde data zoals patiëntverslagen, artsnotities, en onderzoeksliteratuur. Dit kan helpen bij het extraheren van waardevolle inzichten en het automatiseren van documentatieprocessen. Ook kan je met NLP een communicatie bot maken voor vragen.
* Internet of Things (IoT): In de ziekenhuisomgeving kan IoT worden gebruikt voor het verzamelen van real-time data van medische apparatuur, patiëntmonitors en andere sensoren. Deze data kan worden geanalyseerd voor het verbeteren van patiëntenzorg en operationele efficiëntie.

Om het script in productie te zetten zou er een app ontwikkelt kunnen worden die met een gebruiksvriendelijke interface goed gebruikt kan worden. De app zou zo werken:

* Data Import en Weergave: De app kan beginnen met het importeren van de ziekenhuisgegevens. Gebruikers kunnen datasets uploaden of toegang krijgen tot bestaande datasets.
* Analytische Modellen en Visualisaties: De app kan verschillende analytische modellen en visualisaties bieden, zoals de AOC-curve, de Confusion Matrix) en de normale verdeling
* Beveiliging en Privacy: Gezien de gevoeligheid van patiëntengegevens, moet de app voldoen aan alle relevante privacy- en beveiligingsnormen.
* Feature Engineering Resultaten: Gebruikers kunnen inzicht krijgen in de resultaten van feature engineering