# Bijlagen

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

Bijlage 1*: Dataset import*

# 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())

Bijlage 2*: Dataset informatie*

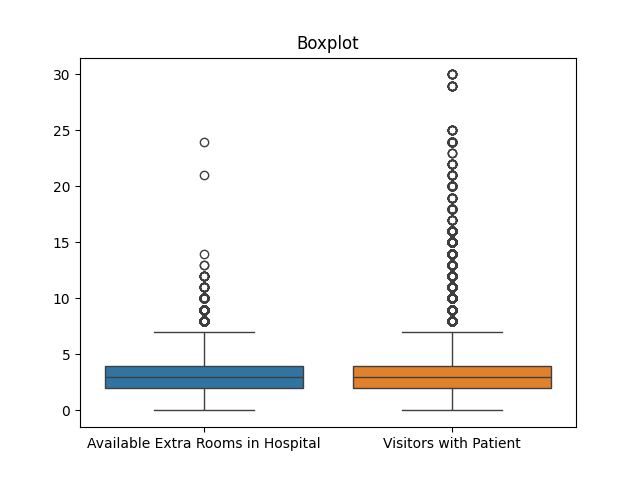
# Drop onnodige kolommen  
print("\nOnnodige kolommen droppen...")  
columns\_to\_drop = ['case\_id', 'patientid']  
traindf = traindf.drop(columns=columns\_to\_drop)  
testdf = testdf.drop(columns=columns\_to\_drop)  
  
# NA waarden in de Bed Grade kolom voor de Train en Test datasets vervangen met 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)

# 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())

Bijlage 3: *Data transformatie*

# 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())

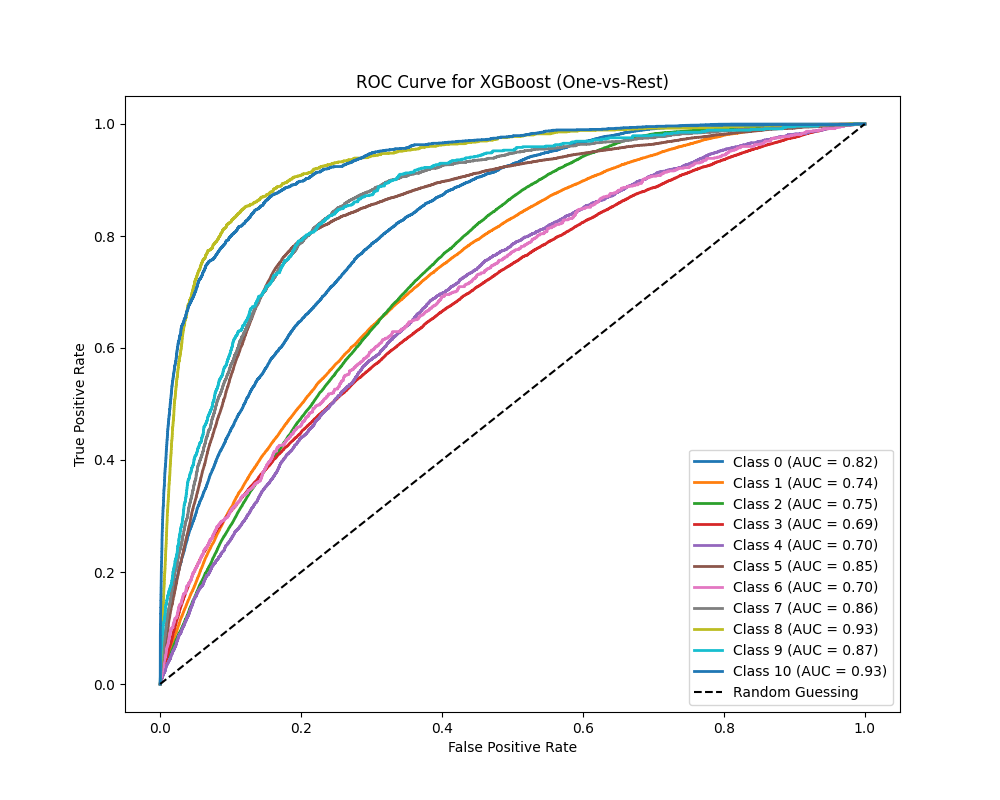
Bijlage 4*: Data transformatie 2*

**

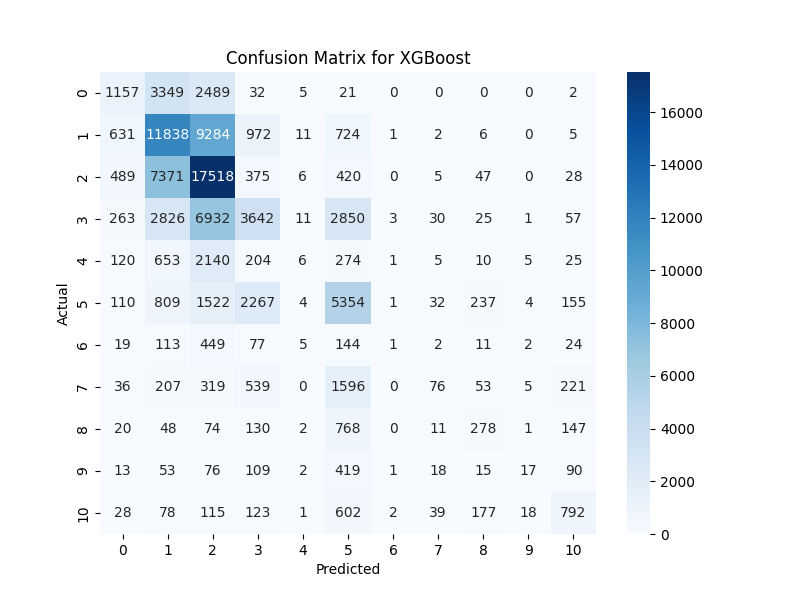
Bijlage 6*: Boxplot*

# Outlier detectie  
def detect\_outliers(df, n, features):  
 outlier\_indices = []  
  
 for col in features:  
 Q1 = np.percentile(df[col], 25)  
 Q3 = np.percentile(df[col], 75)  
 IQR = Q3 - Q1  
 outlier\_step = 1.5 \* IQR  
 outlier\_list\_col = df[(df[col] < Q1 - outlier\_step) | (df[col] > Q3 + outlier\_step)].index  
 outlier\_indices.extend(outlier\_list\_col)  
 outlier\_indices = Counter(outlier\_indices)  
 multiple\_outliers = list(k for k, v in outlier\_indices.items() if v > n)  
 return multiple\_outliers  
  
# outlier detectie voor getallen  
outliers\_to\_drop = detect\_outliers(df, 2, ["Hospital\_code", 'Hospital\_type\_code', 'City\_Code\_Hospital', 'Hospital\_region\_code', 'Available Extra Rooms in Hospital',  
 'Department', 'Ward\_Type', 'Ward\_Facility\_Code', 'Bed Grade', 'City\_Code\_Patient', 'Type of Admission', 'Severity of Illness',  
 'Visitors with Patient', 'Age', 'Admission\_Deposit', 'Stay'])  
# Drop outliers  
df.drop(df.loc[outliers\_to\_drop].index, inplace=True)  
print("\nOutliers verwijderen...")

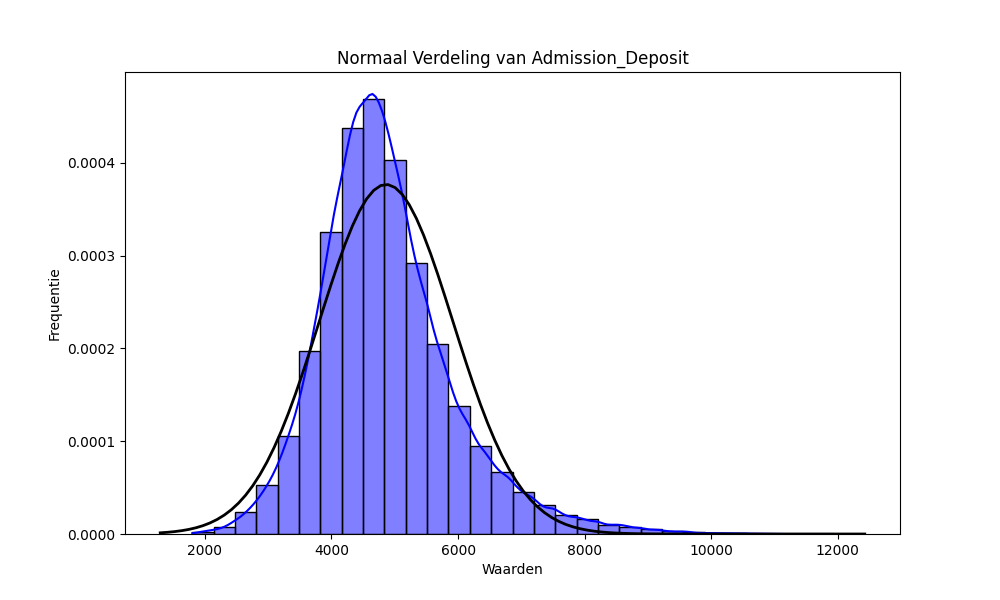
Bijlage 7: *Outlier detectie*

**

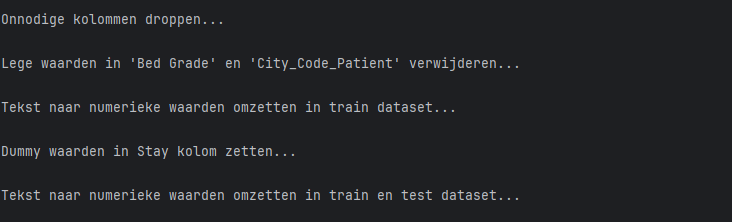
Bijlage 8: *AOC Curve*

**

Bijlage 9: *Confusion Matrix*

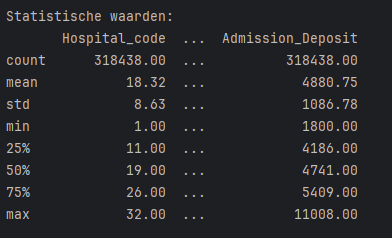
**

Bijlage 10: *Normaal Verdeling*

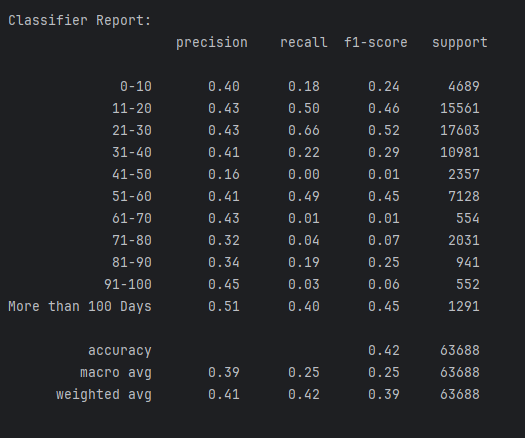


Bijlage 11: *Uitkomsten Feature Engineering*



BIjlage 12: *Model Scores*

Bijlage 13: *Statistische Waarden*



Bijlage 14: *Classifier Report*

# Bronnenlijst

Gupta, R. (2023, March 28). 10 automated EDA tools that will save you hours of work. *Medium*. <https://medium.com/@riteshgupta.ai/10-automated-eda-tools-that-will-save-you-hours-of-work-b3edf3740fc2>

H. Davenport, T. (2006). Competing on Analytics. *Competing on Analytics*. <https://cs.brown.edu/courses/cs295-11/competing.pdf>

Kidu, T. (2023, October 16). Data Analysis with a Single Line of Code Using Advanced Python Libraries: Automate Your EDA and Understand the Data Quickly. *Medium*. <https://medium.com/@ThomTechSavvy/data-analysis-with-a-single-line-of-code-using-advanced-python-libraries-automate-your-eda-and-e1e3fb2ed110>

Redman, T. C. (2013, December). Data’s Credibility Problem. *Enterprisesproject*. <https://enterprisersproject.com/sites/default/files/Data's%20Credibility%20Problem.pdf>