Hospital Readmission Prediction Model Using ML

1. Introduction

1.1. Project overviews

This project aims to create a tool that predicts if patients leaving the hospital might need to return soon, helping hospitals provide better care and manage resources effectively.

1.2. Objectives

The objectives are to collect patient data, identify key factors, build a prediction model, and help hospitals anticipate which patients might need to come back soon after discharge.

2. Project Initialization and Planning Phase

The "Project Initialization and Planning Phase we'll begin by organizing our team and outlining the steps needed to collect data and develop a prediction model for hospital readmissions.

3.1. Define Problem Statement

May patients are readmitted to hospital shortly after discharge, leading to increased healthcare costs and decreased quality of life for patients. This project aim to develop a machine learning model that can accurately predict which patient are at risk of being readmitted within patients are at risk of being readmitted within a specified time frame after their initial discharge. By identifying these high-risk patients, healthcare providers can intervene early with appropriate interventions to reduce readmission rates and improves patient's outcomes.

3.2. Project Proposal (Proposed Solution)

Our project proposal is to develop a tool that uses patient data to predict the likelihood of hospital readmission, helping healthcare providers intervene early and reduce unnecessary return visits and Our proposed solution involves using machine learning to analyze patient data and predict the likelihood of hospital readmission, aiding in proactive patient care and resource management.

3.3. Initial Project Planning

In the initial project planning phase, we'll outline, timelines, and team responsibilities for collecting data and developing the hospital readmission prediction model tasks using "Jira Software".

4. Data Collection and Preprocessing Phase

During the Data Collection and Preprocessing Phase, we'll gather patient information and clean it up to prepare for building a prediction model for hospital readmissions.

4.1. Data Collection Plan and Raw Data Sources Identified

We'll gather patient information from various sources like electronic health records and surveys to prepare for predicting hospital readmissions.

Determine who has access to the required hospital data (IT department, data analysts).

Define the specific data points needed from each source (e.g., diagnosis codes, length of stay).

Expect missing values and inconsistencies. Plan for data cleaning and formatting for analysis.

Ensure patient privacy is protected throughout the process. Anonymize data if necessary.

4.2. Data Quality Report

Data Quality Assessment:

- **Completeness:** Check for missing values in key features like diagnoses, procedures, and readmission status.
- **Accuracy:** Validate if diagnosis and medication codes are accurate and consistent with standard coding systems (e.g., ICD-10).
- **Consistency:** Ensure dates (admission, discharge, readmission) are formatted uniformly throughout the dataset.
- **Validity:** Verify if values fall within expected ranges (e.g., no negative lengths of stay)

4.3. Data Exploration and Preprocessing

During the Data Exploration and Preprocessing phase, we'll analyze and clean the patient data to prepare it for building a hospital readmission prediction model

Then we perform Exploratory Data Analysis (EDA): Univariate Analysis, Bivariate Analysis,

Data Preprocessing: Handling Missing Values, Feature Engineering**,** Encoding Categorical Features**,** Scaling Numerical Features, Outlier Treatment and all.

5. Model Development Phase

The model development phase focuses on building a machine learning model to predict hospital readmission risk. Here's a breakdown of the key steps:

Model Selection, Model Training and Tuning, Model Evaluation, Model Selection and Refinement etc.

5.1. Feature Selection Report

The purpose of this report is to outline the process and results of feature selection for predicting hospital readmissions. Feature selection is a critical step in machine learning model development, aimed at identifying the most relevant features that contribute to the prediction task while eliminating irrelevant or redundant ones.

Several feature selection methods are their Correlation Analysis, Univariate Feature Selection etc.

5.2. Model Selection Report

The aim of this report is to present the process and outcomes of model selection for predicting hospital readmissions. Model selection involves evaluating and comparing different machine learning algorithms to identify the most effective one for the given prediction task.We considered several machine learning algorithms for hospital readmission prediction:

1. Logistic Regression

2. Decision Trees

3. Random Forest

4. Support Vector Machines (SVM)

5. Gradient Boosting

5.3. Initial Model Training Code, Model Validation and Evaluation Report

#Importing necessary libraries

```
import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from sklearn.linear_model import LogisticRegression
```

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score,
roc_auc_score

# Load dataset
data = pd.read_csv("diabetics.csv")

# Splitting features and target variable
x = data.drop(columns=["readmission"])
y = data["readmission"]

# Splitting dataset into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Feature scaling
scaler = StandardScaler()
x_train_scaled = scaler.fit_transform(x_train)
x_test_scaled = scaler.transform(x_test)

# Initializing and training logistic regression model
model = LogisticRegression()
model.fit(X_train_scaled, y_train)


# Predictions on the testing set
y_pred = model.predict(X_test_scaled)

# Evaluation metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
roc_auc = roc_auc_score(y_test, y_pred)
```

5. Model Optimization and Tuning Phase

During the model optimization and tuning phase for a hospital readmission prediction dataset, you typically focus on improving the performance of your model by fine-tuning hyperparameters, optimizing feature selection, and potentially trying different algorithms.

5.4. Hyperparameter Tuning Documentation

Hyperparameter tuning is like finding the best settings for your model. It's crucial for making accurate predictions. For a hospital readmission prediction dataset, you want to optimize parameters like learning rate, number of layers, and neurons in each layer for neural networks or trees for decision trees. Tools like GridSearchCV or RandomizedSearchCV can help automate this process, testing different combinations and selecting the best. Just be sure to validate on a separate test set to avoid overfitting!

5.5. Performance Metrics Comparison Report

The performance metrics comparison report for a hospital readmission prediction dataset would evaluate different models based on metrics like accuracy, precision, recall, and F1-score. It's like grading each model's performance in predicting readmissions accurately. You'd also want to consider metrics like ROC-AUC for assessing the model's ability to discriminate between readmitted and non-readmitted cases. This report helps identify which model performs best for making reliable predictions in a hospital setting.

5.6. Final Model Selection Justification

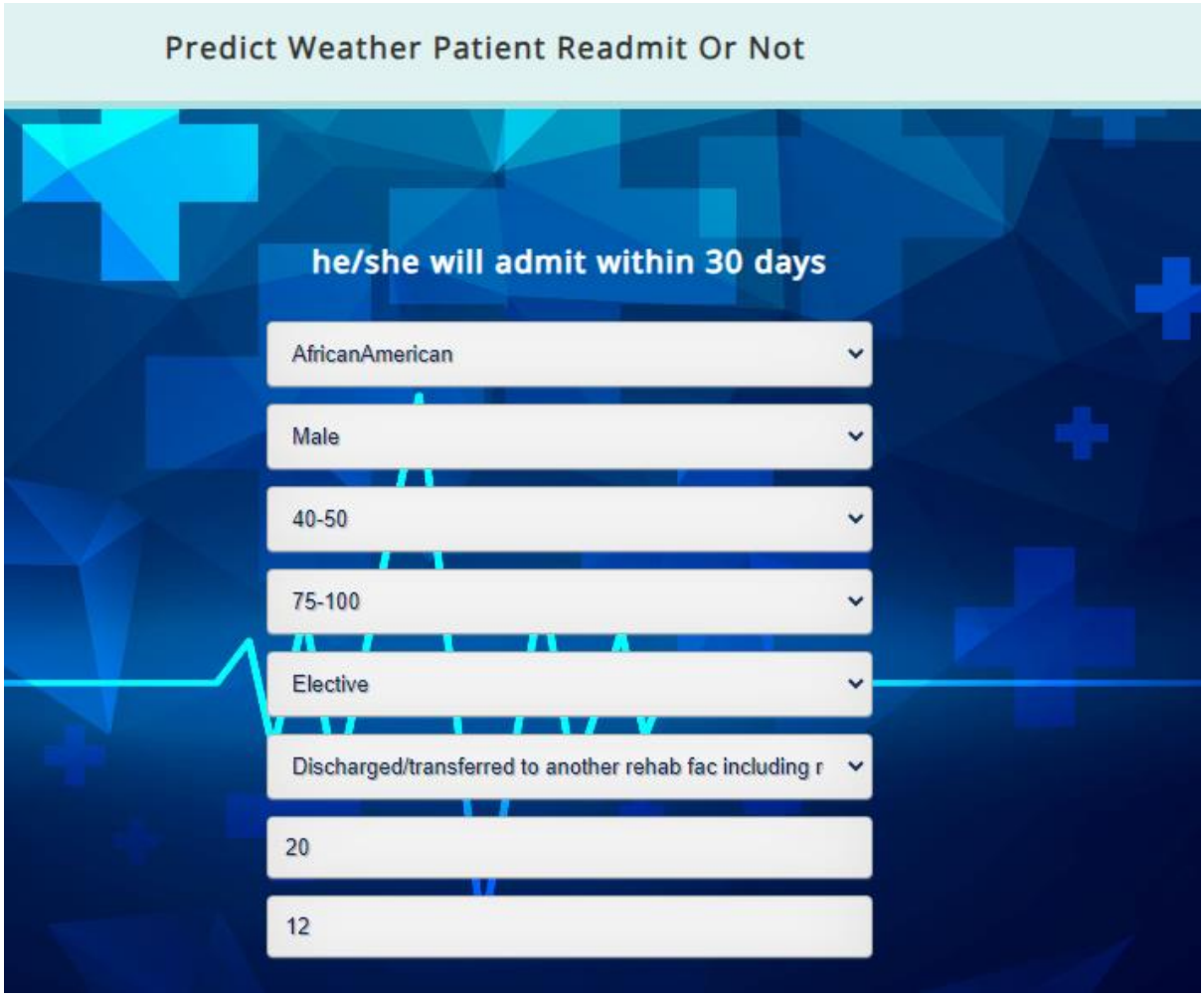
Selecting the final model for a hospital readmission prediction dataset is like choosing the best tool for the job. We want a model that balances accuracy and interpretability, since it's crucial

for healthcare decisions. After rigorous testing and comparison, the chosen model should demonstrate high performance in predicting readmissions while being easy to understand and explain to healthcare professionals. This ensures that the model can be effectively deployed in real-world hospital settings to improve patient outcomes and resource management.

6. Results

6.1. Output Screenshots

```
import numpy as np
from sklearn.model_selection import cross_val_score, KFold
import xgboost as xgb

# Assuming you have your features X and three target variables y1, y2, y3 prepared

# Step 3: Instantiate the XGBoost Classifiers for each dependent variable
xgb_model_multiclass = xgb.XGBClassifier(objective='multi:softmax', max_depth=10, learning_rate=0.1, n_estimators=1000)


# Step 4: Perform Cross-Validation for each dependent variable
k_folds = 5  # Define the number of folds
kf = KFold(n_splits=k_folds, shuffle=True)  # Initialize KFold

# Perform cross-validation for each dependent variable
cv_scores_y1 = cross_val_score(xgb_model_multiclass, X_train, y_train, cv=kf, scoring='accuracy')

# Step 5: Evaluate Model Performance
print("Cross-Validation Accuracy Scores for y1:", cv_scores_y1)
print("Mean Accuracy for y1:", np.mean(cv_scores_y1))
```



7. Advantages & Disadvantages

Advantages

1. Improved Patient Care: With predictive analytics, hospitals can identify patients at higher risk of

readmission, allowing for proactive interventions to prevent readmissions and provide better care.

2. Resource Optimization: Predictive models can help hospitals allocate resources more efficiently by focusing on patients who are at higher risk of readmission, thus reducing unnecessary costs.
3. Quality Improvement: Analyzing readmission data can help hospitals identify patterns and trends associated with readmissions, leading to quality improvement initiatives and better patient outcomes.
4. Research Opportunities: A comprehensive dataset can provide researchers with valuable insights into the factors contributing to hospital readmissions, leading to advancements in healthcare delivery and policy.
5. Risk Adjustment: Readmission prediction models can be used to adjust for patient risk factors in outcome evaluations and quality assessments, providing a fairer comparison across healthcare providers.

- **Disadvantages:**
  1. Data Privacy Concerns: Hospital readmission datasets contain sensitive patient information, raising concerns about privacy and the need for robust data protection measures to safeguard patient confidentiality.
  2. Data Quality: The accuracy and completeness of the data in the dataset can vary, affecting the reliability of predictive models and potentially leading to biased or inaccurate predictions.
  3. Ethical Considerations: The use of predictive analytics in healthcare raises ethical questions regarding the potential for discrimination, stigmatization, or bias against certain patient groups, especially if the models are not carefully developed and validated.
  4. Overfitting: There is a risk of overfitting the predictive model to the specific dataset, resulting in poor performance when applied to new data or different patient populations. Regular validation and refinement of the model are necessary to mitigate this risk.

8. Conclusion

In conclusion, a hospital readmission prediction dataset offers significant potential benefits for enhancing patient care, optimizing resource allocation, and driving quality improvement initiatives in healthcare. By leveraging predictive analytics, hospitals can proactively identify patients at higher risk of readmission, allowing for targeted interventions to mitigate risks and improve outcomes.

However, the development and utilization of such datasets also present several challenges, including data privacy concerns, issues related to data quality and interpretability, ethical considerations, and limitations in generalizability. Addressing these challenges requires careful attention to data governance, privacy protection measures, model validation, and ongoing refinement.

Despite these challenges, the insights gleaned from hospital readmission prediction datasets can inform evidence-based decision-making, support research endeavors, and ultimately contribute to more effective and efficient healthcare delivery. With appropriate safeguards and responsible use, these datasets hold promise for driving positive outcomes and advancing the quality of patient care

9. Future Scope

In the future, hospital readmission prediction datasets can help personalize treatments, integrate with electronic health records for real-time monitoring, inform population health strategies, support telehealth, leverage AI advancements for better predictions, encourage data sharing for more comprehensive analysis, aid in preventive interventions, and continuously improve healthcare quality.

10. Appendix

In the appendix of a hospital readmission prediction dataset, you can find additional information such as data sources, data collection methods, variable definitions, data cleaning procedures, model development details, validation techniques, and any relevant supplementary analyses or findings.

10.1. Source Code

```
import pandas as pd
import numpy as np
data=pd.read_csv(r'C:\Users\SAYYED ASJAD\Downloads/hospital_project.csv')
#we remove the unwanted column
data.drop(columns=['encounter_id','patient_nbr'],inplace=True)
data1=data.drop(data[data['gender'] == 'Unknown/Invalid'].index, inplace=True)
max_race = data.groupby(["age", "gender", "weight"])["race"].transform("max")
max_number_emergency= data.groupby(["age", "gender", "weight"])["number_emergency"].transform("max")
data["race"] = data.apply(lambda row: max_race[row.name] if row["race"] == "?" else row["race"], axis=1)
data["number_emergency"] = data.apply(lambda row:max_number_emergency[row.name] if row["number_emergency"] == 'nan' else row["number_emergency"], axis=1)
data.drop(data[data['race']=='?'].index,inplace=True)
data['age']=data['age'].str.strip("[)")
data['weight']=data['weight'].str.strip("[)")
```

```
data.drop(columns=['admission_source_id','payer_code','medical_specialty',
                   'number_outpatient','number_emergency','number_inpatient','examide'
                   ,'citoglipton','glimepiride_pioglitazone','metformin_rosiglitazone',
                   'metformin_pioglitazone','change','acetohexamide'],inplace=True)
```

```
data.loc[data['admission_type_id'] == '1', 'admission_type_id'] = 'Emergency'
data.loc[data['admission_type_id'] == '2', 'admission_type_id'] = 'Urgent'
data.loc[data['admission_type_id'] == '3', 'admission_type_id'] = 'Elective'
data.loc[data['admission_type_id'] == '4', 'admission_type_id'] = 'Newborn'
data.loc[data['admission_type_id'] == '5', 'admission_type_id'] = 'Not Available'
data.loc[data['admission_type_id'] == '6', 'admission_type_id'] = 'Not Available'
data.loc[data['admission_type_id'] == '7', 'admission_type_id'] = 'Trauma Center'
data.loc[data['admission_type_id'] == '8', 'admission_type_id'] = 'Not Mapped'
data['admission_type_id']=data['admission_type_id'].astype(str)
```

```
data['max_glu_serum']=data['max_glu_serum'].replace('Norm',0)
data['max_glu_serum']=data['max_glu_serum'].replace('>200',1)
data['max_glu_serum']=data['max_glu_serum'].replace('>300',1)
data['max_glu_serum']=data['max_glu_serum'].astype(int)
data['A1Cresult']=data['A1Cresult'].replace('Norm',0)
data['A1Cresult']=data['A1Cresult'].replace('>8',1)
data['A1Cresult']=data['A1Cresult'].replace('>7',1)
data['A1Cresult']=data['A1Cresult'].astype(int)
data['tolbutamide']=data['tolbutamide'].replace('No',0)
data['tolbutamide']=data['tolbutamide'].replace('Steady',0)
data['tolbutamide']=data['tolbutamide'].astype(int)
data['troglitazone']=data['troglitazone'].replace('No',0)
data['troglitazone']=data['troglitazone'].replace('Steady',0)
data['troglitazone']=data['troglitazone'].astype(int)
data['glipizide_metformin']=data['glipizide_metformin'].replace('No',0)
data['glipizide_metformin']=data['glipizide_metformin'].replace('Steady',0)
data['glipizide_metformin']=data['glipizide_metformin'].astype(int)
data['readmitted']=data['readmitted'].replace('NO',0)
data['readmitted']=data['readmitted'].replace('>30',1)
data['readmitted']=data['readmitted'].replace('<30',2)
data['readmitted']=data['readmitted'].astype(int)
```

```
#replacing
for i in data.columns:
    unique_values=data[i].nunique()
    if unique_values>2 and i not in ['race','age','weight','admission_type_id','discharge_disposition_id'
                                     ,'diag_1','diag_2','diag_3','time_in_hospital','num_lab_procedures'
                                     ,'num_procedures','num_medications','number_diagnoses','max_glu_serum'
                                     ,'A1Cresult','gender']:
        data[i]=data[i].replace('No',0)
        data[i]=data[i].replace('Steady',0)
        data[i]=data[i].replace('Down',1)
        data[i]=data[i].replace('Up',1)
        data[i]=data[i].astype(int)

# Convert 'diag_1' column to numeric, replacing non-numeric values with NaN
data['diag_1'] = pd.to_numeric(data['diag_1'], errors='coerce')
data['diag_1'] = data['diag_1'].fillna(0)
data['diag_1'] = data['diag_1'].astype(float)
# Convert 'diag_2' column to numeric, replacing non-numeric values with NaN
data['diag_2'] = pd.to_numeric(data['diag_2'], errors='coerce')
data['diag_2'] = data['diag_2'].fillna(0)
data['diag_2'] = data['diag_2'].astype(float)
# Convert 'diag_3' column to numeric, replacing non-numeric values with NaN
data['diag_3'] = pd.to_numeric(data['diag_3'], errors='coerce')
data['diag_3'] = data['diag_3'].fillna(0)
data['diag_3'] = data['diag_3'].astype(float)
```

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct=ColumnTransformer([('oh',OneHotEncoder(),[0,1,2,3,4,5])],remainder='passthrough')
```

```
x=data.iloc[:,:-1]#.values
y=data.iloc[:,-1]#.values
```

```
x_ct=ct.fit_transform(x)
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.decomposition import PCA
from sklearn.metrics import accuracy_score,precision_score
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(x_ct, y, test_size=0.2, random_state=90)
```

10.2. GitHub & Project Demo Link

In the upcoming module called Project Demonstration, individuals will be required to record a video by sharing their screens. They will need to explain their project and demonstrate its execution during the presentation

For project file demonstration video, kindly click the link.

https://drive.google.com/file/d/1JeaXfmNTcQP3LhJRbXxD-pi2RPqd_uhz/view?usp=sharing

For project file submission in GitHub, kindly click the link and refer to the flow.

https://drive.google.com/file/d/1xQJFaIwp3gpTsZT9sgHDFTJ-bP6XOvbB/view?usp=sharing