Data Collection and Preprocessing Phase

Date	15 March 2024
Team ID	738193
Project Title	Hospital Readmission Prediction Using Machine Learning
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template

Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

Section	Description
Data Overview	Structure of data: columns=50 , rows =101766 Data types : most of data is in form of categorical data.
Univariate Analysis	1.in race column maximum data belongs to caucasian and africanamerican. 2.data has 54708 number of females and 47055 males. 3.data 26068 number of person who belongs to age between 70 to 79 and 2248 persons who belongs to age between 60 to 69. 4.5.7544 is the mean time spending in hospital. 5.mean of number of lab procedure is 43 and max of number of lab procedure id 132. 6.mean number of medication is 16 and max is 81.
Multivariate Analysis	Instead of just focusing on one factor like age or medication, we're considering how various factors such as age, medical history, medications prescribed, and other characteristics all interact to influence the chance of hospital readmission. This type of analysis helps us identify which combination of factors have the strongest impact on readmission rates, helping healthcare providers improve patient care and reduce the likelihood of unnecessary hospital readmissions.
Outliers and Anomalies	Dataset contains outliers we deal with using box plot and replace outlier points with suitable statistics such as if datatype

	is categorical then we replaced with mode of column on three conditions age , gender and race and for numeric we replaced with mean value of that column on 3 conditions age , gender and rce.
Data Preprocessing Code Screenshots	
Loading Data	data=pd.read_csv('hospital_project.csv')
Handling Missing Data	Data does not have missing value but it has some ? mark in each column we deal with these type of data in feature engineering.

Feature Engineering	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)
	#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_1' 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)

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)
Save Processed Data	Our transformed data is saved under data variable.