**SQL Code & Weka**

USE Sepsis

SELECT \* FROM [dbo].[sepsis\_data]

SELECT column\_name

FROM information\_schema.columns

WHERE table\_name = 'sepsis\_data'

SELECT COUNT(DISTINCT Patient\_ID) AS unique\_patients

FROM [dbo].[sepsis\_data]

A screenshot of a computer

Description automatically generated

SELECT TOP 10 \*

FROM [dbo].[sepsis\_data]

--- Calculate percentage of missing values for each column and sort in descending order

WITH missing\_values AS (

SELECT 'HR' AS column\_name, 100.0 \* SUM(CASE WHEN HR IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'O2Sat' AS column\_name, 100.0 \* SUM(CASE WHEN O2Sat IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'Temp' AS column\_name, 100.0 \* SUM(CASE WHEN Temp IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'SBP' AS column\_name, 100.0 \* SUM(CASE WHEN SBP IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'MAP' AS column\_name, 100.0 \* SUM(CASE WHEN MAP IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'DBP' AS column\_name, 100.0 \* SUM(CASE WHEN DBP IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'Resp' AS column\_name, 100.0 \* SUM(CASE WHEN Resp IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'EtCO2' AS column\_name, 100.0 \* SUM(CASE WHEN EtCO2 IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'BaseExcess' AS column\_name, 100.0 \* SUM(CASE WHEN BaseExcess IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'HCO3' AS column\_name, 100.0 \* SUM(CASE WHEN HCO3 IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'FiO2' AS column\_name, 100.0 \* SUM(CASE WHEN FiO2 IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'pH' AS column\_name, 100.0 \* SUM(CASE WHEN pH IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'PaCO2' AS column\_name, 100.0 \* SUM(CASE WHEN PaCO2 IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'SaO2' AS column\_name, 100.0 \* SUM(CASE WHEN SaO2 IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'AST' AS column\_name, 100.0 \* SUM(CASE WHEN AST IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'BUN' AS column\_name, 100.0 \* SUM(CASE WHEN BUN IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'Alkalinephos' AS column\_name, 100.0 \* SUM(CASE WHEN Alkalinephos IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'Calcium' AS column\_name, 100.0 \* SUM(CASE WHEN Calcium IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'Chloride' AS column\_name, 100.0 \* SUM(CASE WHEN Chloride IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'Creatinine' AS column\_name, 100.0 \* SUM(CASE WHEN Creatinine IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'Bilirubin\_direct' AS column\_name, 100.0 \* SUM(CASE WHEN Bilirubin\_direct IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'Glucose' AS column\_name, 100.0 \* SUM(CASE WHEN Glucose IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'Lactate' AS column\_name, 100.0 \* SUM(CASE WHEN Lactate IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'Magnesium' AS column\_name, 100.0 \* SUM(CASE WHEN Magnesium IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'Phosphate' AS column\_name, 100.0 \* SUM(CASE WHEN Phosphate IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'Potassium' AS column\_name, 100.0 \* SUM(CASE WHEN Potassium IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'Bilirubin\_total' AS column\_name, 100.0 \* SUM(CASE WHEN Bilirubin\_total IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'TroponinI' AS column\_name, 100.0 \* SUM(CASE WHEN TroponinI IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'Hct' AS column\_name, 100.0 \* SUM(CASE WHEN Hct IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'Hgb' AS column\_name, 100.0 \* SUM(CASE WHEN Hgb IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'PTT' AS column\_name, 100.0 \* SUM(CASE WHEN PTT IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'WBC' AS column\_name, 100.0 \* SUM(CASE WHEN WBC IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'Fibrinogen' AS column\_name, 100.0 \* SUM(CASE WHEN Fibrinogen IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'Platelets' AS column\_name, 100.0 \* SUM(CASE WHEN Platelets IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'Age' AS column\_name, 100.0 \* SUM(CASE WHEN Age IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'Gender' AS column\_name, 100.0 \* SUM(CASE WHEN Gender IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'Unit1' AS column\_name, 100.0 \* SUM(CASE WHEN Unit1 IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'Unit2' AS column\_name, 100.0 \* SUM(CASE WHEN Unit2 IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'HospAdmTime' AS column\_name, 100.0 \* SUM(CASE WHEN HospAdmTime IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'ICULOS' AS column\_name, 100.0 \* SUM(CASE WHEN ICULOS IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

UNION ALL

SELECT 'SepsisLabel' AS column\_name, 100.0 \* SUM(CASE WHEN SepsisLabel IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM [dbo].[sepsis\_data]

)

SELECT \*

FROM missing\_values

ORDER BY missing\_percentage DESC

---- Selecting only necessary columns for the analysis-------

SELECT Patient\_ID,

[Hour], HR, O2Sat, Temp,

MAP, Resp, [FiO2], [PaCO2],[BUN],[Chloride],

[Creatinine],Glucose,

Hct, Hgb, WBC, Platelets, Age, Gender,

[Unit1], [Unit2],

HospAdmTime, ICULOS, SepsisLabel

INTO dbo. sepsis\_final\_cleaned

FROM [dbo].[sepsis\_data]

select \* from dbo.sepsis\_final\_cleaned

A screenshot of a computer

Description automatically generated

-- Display the columns of the new table

SELECT column\_name

FROM information\_schema.columns

WHERE table\_name = 'sepsis\_final\_cleaned'

A screenshot of a computer

Description automatically generated

WITH missing\_values AS (

SELECT 'Patient\_ID' AS column\_name, 100.0 \* SUM(CASE WHEN Patient\_ID IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM dbo.sepsis\_final\_cleaned

UNION ALL

SELECT '[Hour]' AS column\_name, 100.0 \* SUM(CASE WHEN Hour IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM dbo.sepsis\_final\_cleaned

UNION ALL

SELECT 'HR' AS column\_name, 100.0 \* SUM(CASE WHEN HR IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM dbo.sepsis\_final\_cleaned

UNION ALL

SELECT 'O2Sat' AS column\_name, 100.0 \* SUM(CASE WHEN O2Sat IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM dbo.sepsis\_final\_cleaned

UNION ALL

SELECT 'Temp' AS column\_name, 100.0 \* SUM(CASE WHEN Temp IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM dbo.sepsis\_final\_cleaned

UNION ALL

SELECT 'MAP' AS column\_name, 100.0 \* SUM(CASE WHEN MAP IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM dbo.sepsis\_final\_cleaned

UNION ALL

SELECT 'Resp' AS column\_name, 100.0 \* SUM(CASE WHEN Resp IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM dbo.sepsis\_final\_cleaned

UNION ALL

SELECT 'FiO2' AS column\_name, 100.0 \* SUM(CASE WHEN FiO2 IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM dbo.sepsis\_final\_cleaned

UNION ALL

SELECT 'PaCO2' AS column\_name, 100.0 \* SUM(CASE WHEN PaCO2 IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM dbo.sepsis\_final\_cleaned

UNION ALL

SELECT 'BUN' AS column\_name, 100.0 \* SUM(CASE WHEN BUN IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM dbo.sepsis\_final\_cleaned

UNION ALL

SELECT 'Chloride' AS column\_name, 100.0 \* SUM(CASE WHEN Chloride IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM dbo.sepsis\_final\_cleaned

UNION ALL

SELECT 'Creatinine' AS column\_name, 100.0 \* SUM(CASE WHEN Creatinine IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM dbo.sepsis\_final\_cleaned

UNION ALL

SELECT 'Glucose' AS column\_name, 100.0 \* SUM(CASE WHEN Glucose IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM dbo.sepsis\_final\_cleaned

UNION ALL

SELECT 'Hct' AS column\_name, 100.0 \* SUM(CASE WHEN Hct IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM dbo.sepsis\_final\_cleaned

UNION ALL

SELECT 'Hgb' AS column\_name, 100.0 \* SUM(CASE WHEN Hgb IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM dbo.sepsis\_final\_cleaned

UNION ALL

SELECT 'WBC' AS column\_name, 100.0 \* SUM(CASE WHEN WBC IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM dbo.sepsis\_final\_cleaned

UNION ALL

SELECT 'Platelets' AS column\_name, 100.0 \* SUM(CASE WHEN Platelets IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM dbo.sepsis\_final\_cleaned

UNION ALL

SELECT 'Age' AS column\_name, 100.0 \* SUM(CASE WHEN Age IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM dbo.sepsis\_final\_cleaned

UNION ALL

SELECT 'Gender' AS column\_name, 100.0 \* SUM(CASE WHEN Gender IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM dbo.sepsis\_final\_cleaned

UNION ALL

SELECT '[Unit1]' AS column\_name, 100.0 \* SUM(CASE WHEN Unit1 IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM dbo.sepsis\_final\_cleaned

UNION ALL

SELECT '[Unit2]' AS column\_name, 100.0 \* SUM(CASE WHEN Unit2 IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM dbo.sepsis\_final\_cleaned

UNION ALL

SELECT 'HospAdmTime' AS column\_name, 100.0 \* SUM(CASE WHEN HospAdmTime IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM dbo.sepsis\_final\_cleaned

UNION ALL

SELECT 'ICULOS' AS column\_name, 100.0 \* SUM(CASE WHEN ICULOS IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM dbo.sepsis\_final\_cleaned

UNION ALL

SELECT 'SepsisLabel' AS column\_name, 100.0 \* SUM(CASE WHEN SepsisLabel IS NULL THEN 1 ELSE 0 END) / COUNT(\*) AS missing\_percentage FROM dbo.sepsis\_final\_cleaned

)

SELECT \*

FROM missing\_values

ORDER BY missing\_percentage DESC;

---------As we still have null values, so fill the Nulls with median as it is more appropriate way------

DECLARE @medianHR FLOAT;

-- Calculate median for HR after converting nvarchar to FLOAT

SELECT @medianHR = PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY CAST(HR AS FLOAT)) OVER ()

FROM [dbo].[sepsis\_final\_cleaned] WHERE HR IS NOT NULL AND ISNUMERIC(HR) = 1;

-- Update NULLs with median

UPDATE [dbo].[sepsis\_final\_cleaned]

SET HR = ISNULL(HR, @medianHR);

A screenshot of a computer

Description automatically generated

-----------------

DECLARE @medianO2Sat FLOAT;

-- Calculate median for O2Sat after converting nvarchar to FLOAT, if necessary

SELECT @medianO2Sat = PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY CAST(O2Sat AS FLOAT)) OVER ()

FROM [dbo].[sepsis\_final\_cleaned]

WHERE O2Sat IS NOT NULL AND ISNUMERIC(O2Sat) = 1;

-- Update NULLs with median in O2Sat

UPDATE [dbo].[sepsis\_final\_cleaned]

SET O2Sat = ISNULL(O2Sat, @medianO2Sat);

--------------------

DECLARE @medianTemp FLOAT;

-- Calculate median for Temp after converting nvarchar to FLOAT, if necessary

SELECT @medianTemp = PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY CAST(Temp AS FLOAT)) OVER ()

FROM [dbo].[sepsis\_final\_cleaned]

WHERE Temp IS NOT NULL AND ISNUMERIC(Temp) = 1;

-- Update NULLs with median in Temp

UPDATE [dbo].[sepsis\_final\_cleaned]

SET Temp = ISNULL(Temp, @medianTemp);

----------------

DECLARE @medianMAP FLOAT;

-- Calculate median for MAP after converting nvarchar to FLOAT, if necessary

SELECT @medianMAP = PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY CAST(MAP AS FLOAT)) OVER ()

FROM [dbo].[sepsis\_final\_cleaned]

WHERE MAP IS NOT NULL AND ISNUMERIC(MAP) = 1;

-- Update NULLs with median in MAP

UPDATE [dbo].[sepsis\_final\_cleaned]

SET MAP = ISNULL(MAP, @medianMAP);

----------------------------

DECLARE @medianResp FLOAT;

-- Calculate median for Resp after converting nvarchar to FLOAT, if necessary

SELECT @medianResp = PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY CAST(Resp AS FLOAT)) OVER ()

FROM [dbo].[sepsis\_final\_cleaned]

WHERE Resp IS NOT NULL AND ISNUMERIC(Resp) = 1;

-- Update NULLs with median in Resp

UPDATE [dbo].[sepsis\_final\_cleaned]

SET Resp = ISNULL(Resp, @medianResp);

--------------

DECLARE @medianFiO2 FLOAT;

-- Calculate median for FiO2 after converting nvarchar to FLOAT, if necessary

SELECT @medianFiO2 = PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY CAST(FiO2 AS FLOAT)) OVER ()

FROM [dbo].[sepsis\_final\_cleaned]

WHERE FiO2 IS NOT NULL AND ISNUMERIC(FiO2) = 1;

-- Update NULLs with median in FiO2

UPDATE [dbo].[sepsis\_final\_cleaned]

SET FiO2 = ISNULL(FiO2, @medianFiO2);

-----------------------------

DECLARE @medianPaCO2 FLOAT;

-- Calculate median for PaCO2 after converting nvarchar to FLOAT, if necessary

SELECT @medianPaCO2 = PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY CAST(PaCO2 AS FLOAT)) OVER ()

FROM [dbo].[sepsis\_final\_cleaned]

WHERE PaCO2 IS NOT NULL AND ISNUMERIC(PaCO2) = 1;

-- Update NULLs with median in PaCO2

UPDATE [dbo].[sepsis\_final\_cleaned]

SET PaCO2 = ISNULL(PaCO2, @medianPaCO2);

------------------------------------------------------------

DECLARE @medianBUN FLOAT;

-- Calculate median for BUN after converting nvarchar to FLOAT, if necessary

SELECT @medianBUN = PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY CAST(BUN AS FLOAT)) OVER ()

FROM [dbo].[sepsis\_final\_cleaned]

WHERE BUN IS NOT NULL AND ISNUMERIC(BUN) = 1;

-- Update NULLs with median in BUN

UPDATE [dbo].[sepsis\_final\_cleaned]

SET BUN = ISNULL(BUN, @medianBUN);

---------------------------------

DECLARE @medianChloride FLOAT;

-- Calculate median for Chloride after converting nvarchar to FLOAT, if necessary

SELECT @medianChloride = PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY CAST(Chloride AS FLOAT)) OVER ()

FROM [dbo].[sepsis\_final\_cleaned]

WHERE Chloride IS NOT NULL AND ISNUMERIC(Chloride) = 1;

-- Update NULLs with median in Chloride

UPDATE [dbo].[sepsis\_final\_cleaned]

SET Chloride = ISNULL(Chloride, @medianChloride);

------------------------------

DECLARE @medianCreatinine FLOAT;

-- Calculate median for Creatinine after converting nvarchar to FLOAT, if necessary

SELECT @medianCreatinine = PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY CAST(Creatinine AS FLOAT)) OVER ()

FROM [dbo].[sepsis\_final\_cleaned]

WHERE Creatinine IS NOT NULL AND ISNUMERIC(Creatinine) = 1;

-- Update NULLs with median in Creatinine

UPDATE [dbo].[sepsis\_final\_cleaned]

SET Creatinine = ISNULL(Creatinine, @medianCreatinine);

-------------------------------

DECLARE @medianGlucose FLOAT;

-- Calculate median for Glucose after converting nvarchar to FLOAT, if necessary

SELECT @medianGlucose = PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY CAST(Glucose AS FLOAT)) OVER ()

FROM [dbo].[sepsis\_final\_cleaned]

WHERE Glucose IS NOT NULL AND ISNUMERIC(Glucose) = 1;

-- Update NULLs with median in Glucose

UPDATE [dbo].[sepsis\_final\_cleaned]

SET Glucose = ISNULL(Glucose, @medianGlucose);

------------------------------

DECLARE @medianHct FLOAT;

-- Calculate median for Hct after converting nvarchar to FLOAT, if necessary

SELECT @medianHct = PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY CAST(Hct AS FLOAT)) OVER ()

FROM [dbo].[sepsis\_final\_cleaned]

WHERE Hct IS NOT NULL AND ISNUMERIC(Hct) = 1;

-- Update NULLs with median in Hct

UPDATE [dbo].[sepsis\_final\_cleaned]

SET Hct = ISNULL(Hct, @medianHct);

------------------------------

DECLARE @medianHgb FLOAT;

-- Calculate median for Hgb after converting nvarchar to FLOAT, if necessary

SELECT @medianHgb = PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY CAST(Hgb AS FLOAT)) OVER ()

FROM [dbo].[sepsis\_final\_cleaned]

WHERE Hgb IS NOT NULL AND ISNUMERIC(Hgb) = 1;

-- Update NULLs with median in Hgb

UPDATE [dbo].[sepsis\_final\_cleaned]

SET Hgb = ISNULL(Hgb, @medianHgb);

----------------

DECLARE @medianWBC FLOAT;

-- Calculate median for WBC after converting nvarchar to FLOAT, if necessary

SELECT @medianWBC = PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY CAST(WBC AS FLOAT)) OVER ()

FROM [dbo].[sepsis\_final\_cleaned]

WHERE WBC IS NOT NULL AND ISNUMERIC(WBC) = 1;

-- Update NULLs with median in WBC

UPDATE [dbo].[sepsis\_final\_cleaned]

SET WBC = ISNULL(WBC, @medianWBC);

------------------------------

DECLARE @medianPlatelets FLOAT;

-- Calculate median for Platelets after converting nvarchar to FLOAT, if necessary

SELECT @medianPlatelets = PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY CAST(Platelets AS FLOAT)) OVER ()

FROM [dbo].[sepsis\_final\_cleaned]

WHERE Platelets IS NOT NULL AND ISNUMERIC(Platelets) = 1;

-- Update NULLs with median in Platelets

UPDATE [dbo].[sepsis\_final\_cleaned]

SET Platelets = ISNULL(Platelets, @medianPlatelets);

A screenshot of a computer

Description automatically generated

---------------------------

DECLARE @modeUnit1 VARCHAR(50);

-- Calculate the mode for Unit1

SELECT TOP 1 @modeUnit1 = Unit1

FROM [dbo].[sepsis\_final\_cleaned]

WHERE Unit1 IS NOT NULL

GROUP BY Unit1

ORDER BY COUNT(\*) DESC;

-- Verify the mode value for Unit1

SELECT @modeUnit1 AS ModeUnit1;

-- Update NULLs in Unit1 with the mode value

UPDATE [dbo].[sepsis\_final\_cleaned]

SET Unit1 = ISNULL(Unit1, @modeUnit1);

-- Verify the update for Unit1

SELECT COUNT(\*) AS NullCount\_Unit1

FROM [dbo].[sepsis\_final\_cleaned]

WHERE Unit1 IS NULL;

SELECT Unit1, COUNT(\*) AS Count

FROM [dbo].[sepsis\_final\_cleaned]

GROUP BY Unit1

ORDER BY Count DESC;

------------

DECLARE @modeUnit2 VARCHAR(50);

-- Calculate the mode for Unit2

SELECT TOP 1 @modeUnit2 = Unit2

FROM [dbo].[sepsis\_final\_cleaned]

WHERE Unit2 IS NOT NULL

GROUP BY Unit2

ORDER BY COUNT(\*) DESC;

-- Verify the mode value for Unit2

SELECT @modeUnit2 AS ModeUnit2;

-- Update NULLs in Unit2 with the mode value

UPDATE [dbo].[sepsis\_final\_cleaned]

SET Unit2 = ISNULL(Unit2, @modeUnit2);

-- Verify the update for Unit2

SELECT COUNT(\*) AS NullCount\_Unit2

FROM [dbo].[sepsis\_final\_cleaned]

WHERE Unit2 IS NULL;

SELECT Unit2, COUNT(\*) AS Count FROM [dbo].[sepsis\_final\_cleaned]

GROUP BY Unit2 ORDER BY Count DESC;

A screenshot of a computer

Description automatically generated

----

select \* from dbo.sepsis\_final\_cleaned

A screenshot of a computer

Description automatically generated

----------Calculate the SIRS Score---------------------

-- Step 1: Add columns for individual conditions and the SIRS score

ALTER TABLE dbo.sepsis\_final\_cleaned ADD condition\_temp INT;

ALTER TABLE dbo.sepsis\_final\_cleaned ADD condition\_HR INT;

ALTER TABLE dbo.sepsis\_final\_cleaned ADD condition\_Resp INT;

ALTER TABLE dbo.sepsis\_final\_cleaned ADD condition\_wbc INT;

ALTER TABLE dbo.sepsis\_final\_cleaned ADD SIRS\_Score INT;

-- Step 2: Update the columns with conditions

-- Update condition\_temp with conversion

UPDATE dbo.sepsis\_final\_cleaned

SET condition\_temp = CASE

WHEN TRY\_CAST(Temp AS FLOAT) > 38 OR TRY\_CAST(Temp AS FLOAT) < 36 THEN 1

ELSE 0

END;

-- Update condition\_HR with conversion

UPDATE dbo.sepsis\_final\_cleaned

SET condition\_HR = CASE

WHEN TRY\_CAST(HR AS FLOAT) > 90 THEN 1

ELSE 0

END;

-- Update condition\_Resp with conversion

UPDATE dbo.sepsis\_final\_cleaned

SET condition\_Resp = CASE

WHEN TRY\_CAST(Resp AS FLOAT) > 20 OR TRY\_CAST(PaCO2 AS FLOAT) < 32 THEN 1

ELSE 0

END;

-- Update condition\_wbc with conversion

UPDATE dbo.sepsis\_final\_cleaned

SET condition\_wbc = CASE

WHEN TRY\_CAST(WBC AS FLOAT) > 12000 OR TRY\_CAST(WBC AS FLOAT) < 4000 THEN 1

ELSE 0

END;

-- Step 3: Calculate the SIRS Score

UPDATE dbo.sepsis\_final\_cleaned

SET SIRS\_Score = condition\_temp + condition\_HR + condition\_Resp + condition\_wbc;

-- Step 4: Verify the Results

SELECT SIRS\_Score, COUNT(\*) AS Count

FROM dbo.sepsis\_final\_cleaned

GROUP BY SIRS\_Score

ORDER BY SIRS\_Score

A screenshot of a computer

Description automatically generated

select \* from dbo.sepsis\_final\_cleaned

A screenshot of a computer

Description automatically generated

SELECT

COLUMN\_NAME,

DATA\_TYPE

FROM

INFORMATION\_SCHEMA.COLUMNS

WHERE

TABLE\_NAME = 'sepsis\_data'

--------Divide patients to 3 types - NonSepsis, SepsisAfterAdm , SepsisBeforeAdm in 'dbo.sepsis\_datacleaned'--------------

-- Create a table to store patients with sepsis

SELECT Patient\_ID

INTO #SepticShockPatients

FROM dbo.sepsis\_final\_cleaned

WHERE SepsisLabel = 1

-- Create a table to store patients with sepsis before admission to ICU

SELECT Patient\_ID

INTO #AdmittedWithSepsisPatients

FROM dbo.sepsis\_final\_cleaned

WHERE SepsisLabel = 1 AND Hour = 0

-- Create a table to store patients with sepsis after admission to ICU

SELECT s.Patient\_ID

INTO #SepsisAfterAdmissionPatients

FROM #SepticShockPatients s

LEFT JOIN #AdmittedWithSepsisPatients a ON s.Patient\_ID = a.Patient\_ID

WHERE a.Patient\_ID IS NULL

-- Create a table to store patients without sepsis

SELECT Patient\_ID

INTO #NonSepticPatients

FROM dbo.sepsis\_final\_cleaned

WHERE Patient\_ID NOT IN (SELECT Patient\_ID FROM #SepticShockPatients)

-- Add sepsisType column to the main table

ALTER TABLE dbo.sepsis\_final\_cleaned ADD sepsisType VARCHAR(50)

-- Update sepsisType to SepsisBeforeAdm for patients who were admitted to ICU with sepsis

UPDATE dbo.sepsis\_final\_cleaned

SET sepsisType = 'SepsisBeforeAdm'

WHERE Patient\_ID IN (SELECT Patient\_ID FROM #AdmittedWithSepsisPatients)

-- Update sepsisType to SepsisAfterAdm for patients who contracted sepsis after admission to ICU

UPDATE dbo.sepsis\_final\_cleaned

SET sepsisType = 'SepsisAfterAdm'

WHERE Patient\_ID IN (SELECT Patient\_ID FROM #SepsisAfterAdmissionPatients)

AND sepsisType IS NULL

-- Update sepsisType to NonSepsis for patients who never got Sepsis

UPDATE dbo.sepsis\_final\_cleaned

SET sepsisType = 'NonSepsis'

WHERE Patient\_ID IN (SELECT Patient\_ID FROM #NonSepticPatients)

AND sepsisType IS NULL

-- Check the distribution of sepsisType

SELECT sepsisType, COUNT(\*) AS Count

FROM dbo.sepsis\_final\_cleaned

GROUP BY sepsisType

A screenshot of a computer

Description automatically generated

---------

Exploratory Data Analysis:

use Sepsis

WITH MedianStats AS (

SELECT DISTINCT

SepsisType,

'HR' AS Variable,

PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY TRY\_CAST(HR AS FLOAT)) OVER (PARTITION BY SepsisType) AS Median\_Value

FROM

dbo.sepsis\_final\_cleaned

UNION ALL

SELECT DISTINCT

SepsisType,

'Temp' AS Variable,

PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY TRY\_CAST(Temp AS FLOAT)) OVER (PARTITION BY SepsisType) AS Median\_Value

FROM

dbo.sepsis\_final\_cleaned

UNION ALL

SELECT DISTINCT

SepsisType,

'Resp' AS Variable,

PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY TRY\_CAST(Resp AS FLOAT)) OVER (PARTITION BY SepsisType) AS Median\_Value

FROM

dbo.sepsis\_final\_cleaned

UNION ALL

SELECT DISTINCT

SepsisType,

'WBC' AS Variable,

PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY TRY\_CAST(WBC AS FLOAT)) OVER (PARTITION BY SepsisType) AS Median\_Value

FROM

dbo.sepsis\_final\_cleaned

UNION ALL

SELECT DISTINCT

SepsisType,

'Age' AS Variable,

PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY TRY\_CAST(Age AS FLOAT)) OVER (PARTITION BY SepsisType) AS Median\_Value

FROM

dbo.sepsis\_final\_cleaned

UNION ALL

SELECT DISTINCT

SepsisType,

'ICULOS' AS Variable,

PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY TRY\_CAST(ICULOS AS FLOAT)) OVER (PARTITION BY SepsisType) AS Median\_Value

FROM

dbo.sepsis\_final\_cleaned

),

AggStats AS (

SELECT

SepsisType,

'HR' AS Variable,

AVG(TRY\_CAST(HR AS FLOAT)) AS Mean\_Value,

STDEV(TRY\_CAST(HR AS FLOAT)) AS StdDev\_Value,

COUNT(\*) AS Count\_Value

FROM

dbo.sepsis\_final\_cleaned

GROUP BY SepsisType

UNION ALL

SELECT

SepsisType,

'Temp' AS Variable,

AVG(TRY\_CAST(Temp AS FLOAT)) AS Mean\_Value,

STDEV(TRY\_CAST(Temp AS FLOAT)) AS StdDev\_Value,

COUNT(\*) AS Count\_Value

FROM

dbo.sepsis\_final\_cleaned

GROUP BY SepsisType

UNION ALL

SELECT

SepsisType,

'Resp' AS Variable,

AVG(TRY\_CAST(Resp AS FLOAT)) AS Mean\_Value,

STDEV(TRY\_CAST(Resp AS FLOAT)) AS StdDev\_Value,

COUNT(\*) AS Count\_Value

FROM

dbo.sepsis\_final\_cleaned

GROUP BY SepsisType

UNION ALL

SELECT

SepsisType,

'WBC' AS Variable,

AVG(TRY\_CAST(WBC AS FLOAT)) AS Mean\_Value,

STDEV(TRY\_CAST(WBC AS FLOAT)) AS StdDev\_Value,

COUNT(\*) AS Count\_Value

FROM

dbo.sepsis\_final\_cleaned

GROUP BY SepsisType

UNION ALL

SELECT

SepsisType,

'Age' AS Variable,

AVG(TRY\_CAST(Age AS FLOAT)) AS Mean\_Value,

STDEV(TRY\_CAST(Age AS FLOAT)) AS StdDev\_Value,

COUNT(\*) AS Count\_Value

FROM

dbo.sepsis\_final\_cleaned

GROUP BY SepsisType

UNION ALL

SELECT

SepsisType,

'ICULOS' AS Variable,

AVG(TRY\_CAST(ICULOS AS FLOAT)) AS Mean\_Value,

STDEV(TRY\_CAST(ICULOS AS FLOAT)) AS StdDev\_Value,

COUNT(\*) AS Count\_Value

FROM

dbo.sepsis\_final\_cleaned

GROUP BY SepsisType

)

SELECT

A.SepsisType,

A.Variable,

A.Mean\_Value,

A.StdDev\_Value,

M.Median\_Value,

A.Count\_Value

FROM

AggStats A

JOIN

MedianStats M

ON A.SepsisType = M.SepsisType AND A.Variable = M.Variable

ORDER BY

A.SepsisType, A.Variable;

A screenshot of a computer

Description automatically generated

----------------------------------------------------check for data types and change if needed------------

SELECT

COLUMN\_NAME,

DATA\_TYPE,

CHARACTER\_MAXIMUM\_LENGTH

FROM

INFORMATION\_SCHEMA.COLUMNS

WHERE

TABLE\_NAME = 'sepsis\_final\_cleaned'

----

SELECT

CAST(Patient\_ID AS NVARCHAR(MAX)) AS Patient\_ID,

CAST(Hour AS NVARCHAR(MAX)) AS Hour,

CAST(HR AS NVARCHAR(MAX)) AS HR,

CAST(O2Sat AS NVARCHAR(MAX)) AS O2Sat,

CAST(Temp AS NVARCHAR(MAX)) AS Temp,

CAST(MAP AS NVARCHAR(MAX)) AS MAP,

CAST(Resp AS NVARCHAR(MAX)) AS Resp,

CAST(FiO2 AS NVARCHAR(MAX)) AS FiO2,

CAST(PaCO2 AS NVARCHAR(MAX)) AS PaCO2,

CAST(BUN AS NVARCHAR(MAX)) AS BUN,

CAST(Chloride AS NVARCHAR(MAX)) AS Chloride,

CAST(Creatinine AS NVARCHAR(MAX)) AS Creatinine,

CAST(Glucose AS NVARCHAR(MAX)) AS Glucose,

CAST(Hct AS NVARCHAR(MAX)) AS Hct,

CAST(Hgb AS NVARCHAR(MAX)) AS Hgb,

CAST(WBC AS NVARCHAR(MAX)) AS WBC,

CAST(Platelets AS NVARCHAR(MAX)) AS Platelets,

CAST(Age AS NVARCHAR(MAX)) AS Age,

CAST(Gender AS NVARCHAR(MAX)) AS Gender,

CAST(Unit1 AS NVARCHAR(MAX)) AS Unit1,

CAST(Unit2 AS NVARCHAR(MAX)) AS Unit2,

CAST(HospAdmTime AS NVARCHAR(MAX)) AS HospAdmTime,

CAST(ICULOS AS NVARCHAR(MAX)) AS ICULOS,

CAST(SepsisLabel AS NVARCHAR(MAX)) AS SepsisLabel,

condition\_temp, -- No conversion needed for int columns

condition\_HR,

condition\_Resp,

condition\_wbc,

SIRS\_Score,

CAST(sepsisType AS NVARCHAR(MAX)) AS sepsisType -- Convert to NVARCHAR(MAX)

INTO

sepsis\_final\_cleaned\_export

FROM

sepsis\_final\_cleaned;

------------------

select \* from dbo.sepsis\_final\_cleaned\_export

SELECT

COLUMN\_NAME,

DATA\_TYPE

FROM

INFORMATION\_SCHEMA.COLUMNS

WHERE

TABLE\_NAME = 'sepsis\_final\_cleaned\_export';

SELECT TOP 100000 \*

INTO dbo.sepsis\_sub

FROM dbo.sepsis\_final\_cleaned\_export

select \* from dbo.sepsis\_sub

A screenshot of a computer

Description automatically generated

**Weka:**

Loaded data to Weka

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**RandomForest Algorithm:**A screenshot of a computer

Description automatically generated

**Logistic Regression**:

A screenshot of a computer

Description automatically generated

**NaiveBayes:**

A screenshot of a computer

Description automatically generated

The above classification methods are done using class: SepsisLabel i.e. 0 or 1

Attribute selection – GainRatioAttributeEval

A screenshot of a computer program

Description automatically generated

Now in the weka Experiment Environment I compared all the three models I did above, here is the Screenshot:

A screenshot of a computer

Description automatically generated

Additionally, I also experimented with classification method using class -SepsisType.

Here is a screenshot of RandomForest Model:

A screenshot of a computer

Description automatically generated