**SQL EDA**

***Business insights Report: Before & After data* preprocessing in my SQL**

INTRODUCTION:

Machine failure prediction is the most common issue in among all the industries in the world. In each company has so many machines and in each unit, every day generates huge amount of data. Good sql exploratory data analysis is important to get clear insights from this vast amount of data. Our SQL EDA gives insights into performance of the teams and players over a period of time.

Before Data Preprocessing:

Data Source:

The data source was extracted to get insights into machine performance consisted of more than 2500 records which covers information related games played form around two years.

As the data is presented at a machine we had to create a separate data connection to aggregate the data at assembly line level. Overall we used both the raw data and a custom SQL output.

Data is extracted from our aispry website at project session at data set:.., which contains statistics and historical data. Machine data will be collected to use in the business insights.

Data Quality Assessment:

* I identify the missing values, duplicates and outliers – date, hydraulic pressure, machine-id, coolent-pressure, air\_system\_pressuure, Coolent\_temparature, Hydraulic\_temparature, spindle bearing temp, spindle speed , spindle vibration, tool vibration, voltage, Torque cutting, downtime.
* I identify the potential problems that could affect analysis and decision-making - downtime
* I identify the inconsistencies – downtime
* Hot encoding (categorical to int) – assembly\_line, downtime.

**Data Exploration:**

I investigate the downtime column to understand the reasons for machine failure prediction, I analyze the frequency and duration of downtime events to improve machine availability . parameter like cutting force(kn) can be correlated with product quality. If cutting force is greater than 30 in 3(shop floor L3) machine has no failure rate. Our main objective is to minimize the machine failure. Use this historical data to predict &develop the predictive maintenance models. Predict when machines might fail data to schedule maintenance proactively. Monitor the coolent pressure & temperature to ensure that the machining processes remains efficient and that tools are adequately cooled. Analyze the power consumption data(voltage & torque) to identify the opportunities for energy savings and sustainability improvements.

These insights can help in optimizing operations, improving product quality, reducing downtime, and enhancing in a manufacturing or industrial setting. It’s important to conduct in-depth data analysis, SQL EDA, and potentially machine learning modelling to extract actionable insights.

**Data preprocessing :**

**Data cleaning:**

* ***Handling missing values***:- in all columns except ‘assembly line no’ (filing integer values with mean, median & categorical values with mode.
* ***Removing duplicates***:- identify the false values and replace with correct values.
* ***Dealing with outliers:-*** anomalies like present in downtime.
* ***Standardization or normalization:-*** standardization is useful when you want to maintain the shape of the data distribution and handle outliers better, while normalization scales data to a fixed range and can be more sensitive to outliers.
* ***Encoding categorical values:-*** creating new variables with existing columns using select CASE ***ex:*** select case when column name >threshold then ‘high’ else ‘low’ end as new\_column from your\_table;
* ***Feature engineering:-*** binning, aggregation, feature scaling,date &time time features.

***Data transformation:- Hot encoding(categorical variables to integer valuesd) at assembly line no & down time. It improves the data’s suitability for analysis.***

***SQL CODE:-***

The following sql code is used for data preprocessing

I used mysql xamp so need to create table, I just change the required structure…

Select \* from new\_machine\_data;

Select distinct \* form new\_machine\_data;

SELECT \* case WHEN `Assembly\_Line\_No`= 'Shopfloor-L1' THEN 1 else 0 END as is\_Shopfloor\_L1,case WHEN `Assembly\_Line\_No`= 'Shopfloor-L2' THEN 2 else 0 END as is\_Shopfloor\_L2,case WHEN `Assembly\_Line\_No`= 'Shopfloor-L3' THEN 3 else 0 END as is\_Shopfloor\_L3 FROM `new\_machine\_downtime`;

Commit;

Alter table new\_machine\_data modify column date DATE;

Commit;

Update new\_machine\_data set machine = ‘Makino-L1-Unit1-2013 where `Assembly\_Line\_No`=1 form new\_machine\_data;

Update new\_machine\_data set machine = ‘Makino-L2-Unit1-2015 where `Assembly\_Line\_No`=2 form new\_machine\_data;

Update new\_machine\_data set machine = ‘Makino-L3-Unit1-2015 where `Assembly\_Line\_No`=3 form new\_machine\_data;

Commit;

Update new\_machine\_data set date = str\_to\_date(date, ‘%dd/%mm/%yyyy’) where str\_to\_date(date, ‘%dd/%mm/%yyyy’) is null;

Commit;

SELECT COUNT(\*) AS NULL\_COUNT\_date FROM new\_machine\_downtime WHERE date IS NULL;

Update new\_machine\_data as t1

Set date = ( select max(date) from new\_machine\_data as t2

Where t2.date<t1.date and t2.date is not null )

Where date is null;

Update new\_machine\_data as t1

Set date = ( select min(date) from new\_machine\_data as t2

Where t2.date<t1.date and t2.date is not null )

Where date is null;

Commit;

SELECT COUNT(\*) AS NULL\_COUNT\_Machine\_id FROM new\_machine\_downtime WHERE Machine\_id IS NULL;

Alter table new\_machine\_data modify column Machine\_id varchar2(30) not null;

Commit;

SELECT COUNT(\*) AS NULL\_COUNT\_Assembly\_line\_no FROM new\_machine\_downtime WHERE Assembly\_line\_no IS NULL;

Update new\_machine\_data set machine\_id = concat(‘makino-L’, Assembly\_line\_no,’-Unit1-2013’) where (select \* from new\_machine\_data where Assembly\_line\_no =1) machine\_id is null;

Update new\_machine\_data set machine\_id = concat(‘makino-L’, Assembly\_line\_no,’-Unit1-2015’) where (select \* from new\_machine\_data where Assembly\_line\_no =in(2,3)) machine\_id is null;

Commit;

Alter table new\_machine\_data modify column Assemby\_line\_no varchar(20) not null;

Commit;

SELECT COUNT(\*) AS NULL\_COUNT\_Hydraulic\_pressure FROM new\_machine\_downtime WHERE Hydraulic\_pressure IS NULL;

Update new\_machine\_data set Hydraulic\_pressure = ( select avg(Hydraulic\_pressure) from new\_machine\_data where Hydaraulic\_pressure is not null) where Hydraulic\_pressure is null;

Alter table new\_machine\_data modify column Hydaraulic\_pressure FLOAT(10) NOTNULL;

Commit;

SELECT COUNT(\*) AS NULL\_COUNT\_Coolent\_pressure FROM new\_machine\_downtime WHERE Coolent\_pressure IS NULL;

Update new\_machine\_data set Coolent\_pressure = ( select avg(Coolent\_pressure) from new\_machine\_data where Coolent\_pressure is not null) where Coolent\_pressure is null;

Alter table new\_machine\_data modify column Coolent\_pressure FLOAT(10) NOTNULL;

Commit;

SELECT COUNT(\*) AS NULL\_COUNT\_Air\_System\_pressure FROM new\_machine\_downtime WHERE Air\_System\_pressure IS NULL;

Update new\_machine\_data set Air\_System\_pressure = ( select avg(Air\_System \_pressure) from new\_machine\_data where Air\_System\_pressure is not null) where Air\_System\_pressure is null;

Alter table new\_machine\_data modify column Air\_System\_pressure FLOAT(10) NOTNULL;

Commit;

SELECT COUNT(\*) AS NULL\_COUNT\_Coolent\_Temperature FROM new\_machine\_downtime WHERE Coolent\_Temperature IS NULL;

Update new\_machine\_data set Coolent\_Temperature= ( select avg(Coolent \_ Temperature) from new\_machine\_data where Coolent\_Temperature is not null) where Coolent\_Temperature is null;

Alter table new\_machine\_data modify column Coolent\_Temperature FLOAT(4) NOTNULL;

Commit;

SELECT COUNT(\*) AS NULL\_COUNT\_Hydraulic\_Oil\_Temperature FROM new\_machine\_downtime WHERE Hydraulic\_Oil\_Temperature IS NULL;

Update new\_machine\_data set Hydraulic\_Oil\_Temperature= ( select avg(Hydraulic\_Oil\_Temperature) from new\_machine\_data where Hydraulic\_Oil\_Temperature is not null) where Hydraulic\_Oil\_Temperature is null;

Alter table new\_machine\_data modify column Hydraulic\_Oil\_Temperature FLOAT(4) NOTNULL;

Commit;

SELECT COUNT(\*) AS NULL\_COUNT\_Spindle\_Bearing\_Temperature FROM new\_machine\_downtime WHERE Spindle\_Bearing\_Temperature IS NULL;

Update new\_machine\_data set Spindle\_Bearing\_Temperature= ( select avg(Spindle\_Bearing\_Temperature) from new\_machine\_data where Spindle\_Bearing\_Temperature is not null) where Spindle\_Bearing\_Temperature is null;

Alter table new\_machine\_data modify column Spindle\_Bearing\_Temperature FLOAT(4) NOTNULL;

Commit;

SELECT COUNT(\*) AS NULL\_COUNT\_Spindle\_Vibration FROM new\_machine\_downtime WHERE Spindle\_Vibration IS NULL;

Update new\_machine\_data set Spindle\_Vibration= ( select avg(Spindle\_Vibration) from new\_machine\_data where Spindle\_Vibration is not null) where Spindle\_Vibration is null;

Alter table new\_machine\_data modify column Spindle\_Vibration FLOAT(6) NOTNULL;

Commit;

SELECT COUNT(\*) AS NULL\_COUNT\_Tool\_Vibration FROM new\_machine\_downtime WHERE Tool\_Vibration IS NULL;

Update new\_machine\_data set Tool\_Vibration= ( select avg(Tool\_Vibration) from new\_machine\_data where Tool\_Vibration is not null) where Tool\_Vibration is null;

Alter table new\_machine\_data modify column Tool\_Vibration FLOAT (6) NOTNULL;

Commit;

SELECT COUNT(\*) AS NULL\_COUNT\_Spindle\_Speed FROM new\_machine\_downtime WHERE Spindle\_Speed IS NULL;

Update new\_machine\_data set Spindle\_Speed= ( select avg(Spindle\_Speed) from new\_machine\_data where Spindle\_Speed is not null) where Spindle\_Speed is null;

Alter table new\_machine\_data modify column Spindle\_Speed INT(6) NOTNULL;

Commit;

SELECT COUNT(\*) AS 'NULL\_COUNT\_ Voltage FROM `new\_machine\_downtime` WHERE ‘Voltage' IS NULL;

Update new\_machine\_data set Voltage= ( select avg(Voltage) from new\_machine\_data where Voltage is not null) where Voltage is null;

Alter table new\_machine\_data modify column Voltage INT(3)NOTNULL;

Commit;

SELECT COUNT(\*) AS 'NULL\_COUNT\_ Torque FROM `new\_machine\_downtime` WHERE ‘Torque ' IS NULL;

Update new\_machine\_data set Torque= ( select avg(Torque) from new\_machine\_data where Torque is not null) where Torque is null;

Alter table new\_machine\_data modify column Torque FLOAT(9)NOTNULL;

Commit;

SELECT COUNT(\*) AS 'NULL\_COUNT\_ Cutting' FROM `new\_machine\_downtime` WHERE ‘Cutting' IS NULL;

Update new\_machine\_data set Cutting= ( select avg(Cutting) from new\_machine\_data where Cutting is not null) where Cutting is null;

Alter table new\_machine\_data modify column Cutting FLOAT(3)NOTNULL;

Commit;

SELECT COUNT(\*) AS 'NULL\_COUNT' FROM `new\_machine\_downtime` WHERE 'Downtime' IS NULL;

Update new\_machine\_data

Set Downtime = ( select mode\_value form (select Downtime as mode\_value, count(\*) as frequency from (select \* from new\_machine\_downtime where Assembly\_Line\_no = 1)where Downtime is not null Group By Downtime Order By frequency DESC Limit 1) as mode\_subquery

Where Downtime is null ;

Commit;

Update new\_machine\_data

Set Downtime = ( select mode\_value form (select Downtime as mode\_value, count(\*) as frequency from (select \* from new\_machine\_downtime where Assembly\_Line\_no = 2)where Downtime is not null Group By Downtime Order By frequency DESC Limit 1) as mode\_subquery

Where Downtime is null ;

Commit;

Update new\_machine\_data

Set Downtime = ( select mode\_value form (select Downtime as mode\_value, count(\*) as frequency from (select \* from new\_machine\_downtime where Assembly\_Line\_no = 3)where Downtime is not null Group By Downtime Order By frequency DESC Limit 1) as mode\_subquery

Where Downtime is null ;

Alter table new\_machine\_data modify column Downtime varchar(20) NOTNULL;

Commit;

SELECT\* FROM 'new\_machine\_downtime' where month(`Date`)=1;

***After data preprocessing:-***

***Data quality improvement:-***

***What is the trend in the performance of the each assembly line?***

* The assembly line give the information about machine working with respect to date and whether they have machine failure rate or no machine failure rate.
* This sql eda also gives the information on what the condition on that particular date for machine.
* The data also compared against the season average to get the insights on where they stand compared to other machines on the same unit.

***Against which machine is strong and against which they are weak?***

* the sql eda is prepared here will gave the information on the % of machine\_failure and no\_machine\_failure against each of the season. Sort by date by no\_machine\_failure to get a quick view on against which machine is strong.

***What are machine failures and noi machine failures of the season or year***

* this sql eda will give information on by how many times machine fails.
* This can help in preparing statergy when running machine with these seasons and con improve.

***Data Exploration(post -processing):***

* It is a critical step to gain insights into our preprocessed downtime data and ensure that the preprocessing steps have had the desired effect.
* Calculate the summary statistics for relevant columns , such as mean median, standard deviation, minimum, & maximum. This helps in finding the central tendencies and variability of data.
* Continue or reevaluate outlier detection to ensure that preprocessing steps effectively handled outliers
* Check for class imbalance on classification problems.
* Our data involves time series, performs type cast string to date.
* For full quality assessment to ensure that preprocessing didn’t iintroduce any new issues to the table structure.
* Check for any missing values

***Business Insights:***

* + After data preprocessing for machine downtime can provide operational efficiency, reduce downtime, and optimize maintenance strategies.
  + Most of the machine failures occurred in 2nd month.
  + Most of the no machine failures occurred in 1th month.
  + More frequently machine failures are increased with day by day.
  + Voltage & torque are causes into most frequently.
  + High temperatures which causes with respect to season and with high cutting speed which causes the machine downtime.
  + I analyze that there is a less duration of downtime in 3rd month.
  + More machine failures occur at 2nd month due to low tool vibration or high spindle speed.
  + Due to this machine failures it impacts on the financial, including lost production, maintenance costs and potential revenue loses. This insights can help prioritize preventive measures.
  + So, our main objective is to be reduce the machine failures or machine downtime for overcome the financial loses.
  + Unplanned downtime leads to financial lose. So, we maintain the effectiveness of activities by comparing downtime. This pattern is indicating the reducing the unplanned machine downtime.
  + Comparing the machine metrics like vibration, temperature, pressure and their correlation with downtime events. It helps to find the thresholds values for finding exact missing or outliers in machine downtime.
  + Minimize downtime during critical production periods.
  + Based on the resource availability we maintain the scheduled optimization.
  + Efficiently by analyzing which machines are most prone to downtime for allocate resources.
  + The performance for identify the areas for improvement the data with data perprocessing in sql.
  + Perform cost-benefit analyses of potential solutions such as upgrading machinery, implementing the new maintenance procedures, or investing in predictive maintenance technology.
  + Regularly reviewing downtime data and implementing changes based on insights for better improvements in operational efficiency and cost savings.