

EduBridge assessment solutions

NAME : P. JAHNAVI RACHEL

ROLL NO. : 2000520040 (KLU)

TOPIC : SQL

The Faculty needs a section-wise Number of candidates who have secured more than or equal to 75 marks in the Semester Exam.

- We can achieve this by using MySQL.
- Here are the steps to be followed:

Step 1: Create a Database and use it by giving the below commands in MySQL–

```
mysql> CREATE DATABASE database_name;  
mysql> USE database_name;
```

Step 2: Create a Table

```
CREATE TABLE table_name(  
  
    column_definition1,  
    column_definition2,  
  
    .....  
    table_constraints  
  
);
```

```
Select MySQL 8.0 Command Line Client  
Enter password: ****  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 11  
Server version: 8.0.27 MySQL Community Server - GPL  
  
Copyright (c) 2000, 2021, Oracle and/or its affiliates.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> create database dab;  
Query OK, 1 row affected (0.49 sec)  
  
mysql> use dab;  
Database changed  
mysql> create table tab(EnrollmentNo int(10),StudentName varchar(10),Section char(10),StudentId int(10),Marks int(100));  
Query OK, 0 rows affected, 3 warnings (3.41 sec)  
  
mysql> desc tab;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type      | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| EnrollmentNo | int       | YES  |     | NULL    |       |  
| StudentName  | varchar(10) | YES  |     | NULL    |       |  
| Section      | char(10)   | YES  |     | NULL    |       |  
| StudentId    | int       | YES  |     | NULL    |       |  
| Marks        | int       | YES  |     | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
5 rows in set (0.72 sec)
```

Step 3: Insert data values into table


```
INSERT INTO table_name VALUES (value1, value2, value3, ...);
```

Step 4: Display the table with values

```
SELECT * FROM table_name;
```

Step 5: Use the following command to get section-wise Number of candidates who have secured more than or equal to 75 marks in the Semester Exam

```
select column_name, count(DISTINCT(column_name)) From table_name GROUP BY column_name;
```

 Select MySQL 8.0 Command Line Client

```
mysql> INSERT INTO tab VALUES(1, 'Tim', 'A', 1, 70),  
-> (2, 'Jim', 'A', 2, 75),  
-> (3, 'Kim', 'B', 3, 65),  
-> (4, 'Tom', 'B', 4, 77),  
-> (5, 'John', 'C', 5, 60),  
-> (6, 'Joe', 'C', 1, 82),  
-> (7, 'James', 'B', 2, 76),  
-> (8, 'Henry', 'C', 5, 68),  
-> (9, 'Matt', 'B', 3, 71),  
-> (10, 'Paul', 'A', 4, 79);
```

```
Query OK, 10 rows affected (0.30 sec)  
Records: 10 Duplicates: 0 Warnings: 0
```

```
mysql> select * from tab;
```

EnrollmentNo	StudentName	Section	StudentId	Marks
1	Tim	A	1	70
2	Jim	A	2	75
3	Kim	B	3	65
4	Tom	B	4	77
5	John	C	5	60
6	Joe	C	1	82
7	James	B	2	76
8	Henry	C	5	68
9	Matt	B	3	71
10	Paul	A	4	79

```
10 rows in set (0.00 sec)
```

```
mysql> select Section, count(DISTINCT(Marks)) From tab GROUP BY Section;
```

Section	count(DISTINCT(Marks))
A	3
B	4
C	3

```
3 rows in set (0.00 sec)
```

```
mysql> _
```

Topic : Tableau

Arun has the following data of Employees in CSV format

```
Emp_name, Id,Salary
Ravish,10,1000
Suresh,101,20000
Priya,1010,50000
Neha,10101,70000
Nitin,1101,15000
```

To make all Employee Id (Id) a 7-digit number in Tableau we need to follow the below steps :

STEP 1: We need to upload the given data CSV file in Tableau Workbook.

STEP 2: After uploading we can see the dimensions at the left side of the tableau workbook.

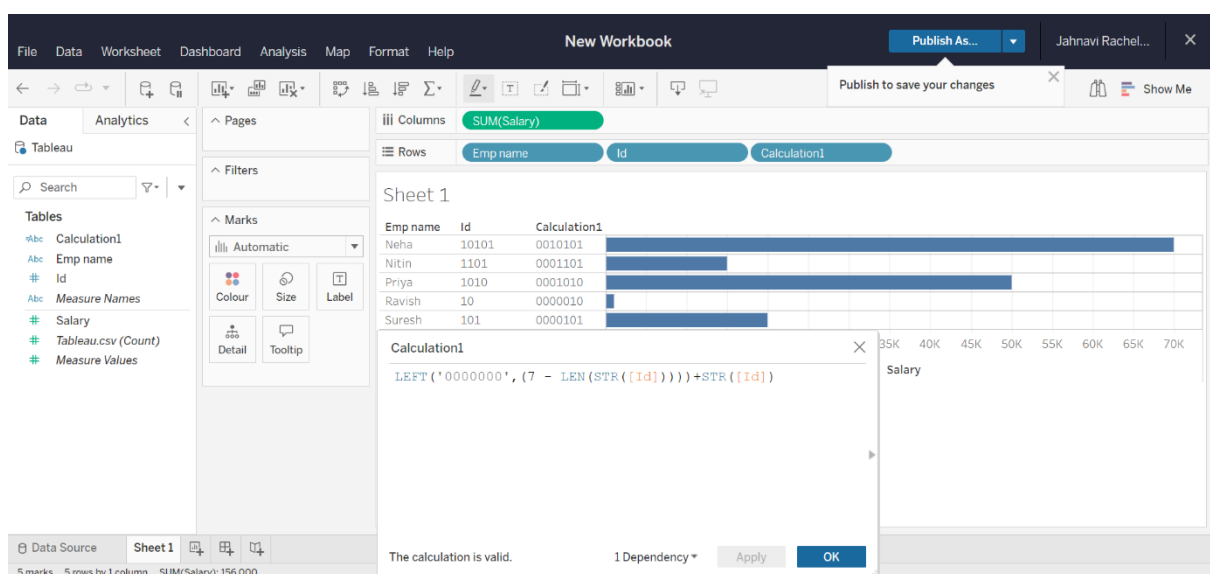
STEP 3: We need to drag dimensions to the 'Rows' (It's not mandatory to choose row or column we can choose any one based on our convenience and understanding).

STEP 5: I dragged the 'Salary' dimension to the 'Columns'.

STEP 6: Now click on 'Analysis' and choose "create calculated field" and write the below code :

```
LEFT('0000000',(7-LEN(STR([Id]))) + STR([Id])
```

STEP 7: Click on 'Apply' and 'OK'.



Topic: Excel

To populate the Duplicate name field for names that occurs more than once we can use a formula to display them.

STEP 1: Create two columns as shown in the question

STEP 2: Click on first cell of 2nd column which is 'Duplicate name' and enter the below formula in the formula bar.

```
=IF(COUNTIF($A$2:$A$17, $A2)>1, A2, "")
```

STEP 3: Click on SHIFT + CTRL + ENTER

STEP 4: Then select B2 and drag the fill handle to copy the formula down to other cells.

	A	B
1	Customer name	Duplicate name
2	Kapil khatri	Kapil khatri
3	Arti Ahuja	Arti Ahuja
4	Eshank sharma	Eshank sharma
5	Amit kumar	Amit kumar
6	Kapil khatri	Kapil khatri
7	Raj Sharma	
8	Sunil Yadav	
9	Eshank sharma	Eshank sharma
10	Swati Singh	
11	Animesh verma	
12	Mohit Jain	
13	Arti Ahuja	Arti Ahuja
14	Ashutosh Mahajan	
15	Akshay Rathod	
16	Harmeet kaur	
17	Amit kumar	Amit kumar

As you can see in the screenshot above, the formula the duplicate values.

Topic: Machine Learning

To classify fetal health to avoid the abnormalities for the child and mother while giving birth.

STEP 1: import libraries as shown below

```
[7] # Importing Libraries

import numpy as np
import pandas as pd
import scipy.stats as stats
from matplotlib import pyplot as plt
%matplotlib inline
from sklearn.preprocessing import StandardScaler
```

STEP 2 : Read the path of the file

```
[8] # Loading dataset
df = pd.read_csv('/content/train.csv')
```

df

	baseline value	accelerations	fetal_movement	uterine_contractions	light_decelerations	severe_decelerations	prolongued_decelerations	abnormal_shor
0	142.0	0.000	0.000	0.007	0.000	0.0	0.000	
1	122.0	0.000	0.000	0.006	0.002	0.0	0.000	
2	129.0	0.005	0.003	0.001	0.000	0.0	0.000	
3	136.0	0.006	0.000	0.008	0.000	0.0	0.000	
4	144.0	0.000	0.000	0.006	0.000	0.0	0.000	
...
1695	129.0	0.000	0.001	0.006	0.008	0.0	0.002	
1696	142.0	0.001	0.003	0.001	0.002	0.0	0.000	
1697	147.0	0.013	0.000	0.010	0.000	0.0	0.000	
1698	130.0	0.003	0.000	0.007	0.000	0.0	0.000	

3s completed at 22:28

STEP 3: Declaring the objectives

```
[11] X = pd.DataFrame(df.iloc[:, :-1])
```

X

	baseline value	accelerations	fetal_movement	uterine_contractions	light_decelerations	severe_decelerations	prolongued_decelerations	abnormal_shor
0	142.0	0.000	0.000	0.007	0.000	0.0	0.000	
1	122.0	0.000	0.000	0.006	0.002	0.0	0.000	
2	129.0	0.005	0.003	0.001	0.000	0.0	0.000	
3	136.0	0.006	0.000	0.008	0.000	0.0	0.000	
4	144.0	0.000	0.000	0.006	0.000	0.0	0.000	
...
1695	129.0	0.000	0.001	0.006	0.008	0.0	0.002	
1696	142.0	0.001	0.003	0.001	0.002	0.0	0.000	
1697	147.0	0.013	0.000	0.010	0.000	0.0	0.000	
1698	130.0	0.003	0.000	0.007	0.000	0.0	0.000	

3s completed at 22:28

We can make the below output of 'fata_health' column into CSV

```
[13] Y = pd.DataFrame(df.iloc[:, -1:])
```

Y

	fetal_health
0	1.0
1	1.0
2	1.0
3	1.0
4	1.0
...	...
1695	3.0
1696	1.0
1697	1.0
1698	1.0
1699	1.0

STEP 4: Data pre-processing by handling missing values

```
# handling missing values
df.isnull().sum() #No null values
```

baseline_value	0
accelerations	0
fetal_movement	0
uterine_contractions	0
light_decelerations	0
severe_decelerations	0
prolongued_decelerations	0
abnormal_short_term_variability	0
mean_value_of_short_term_variability	0
percentage_of_time_with_abnormal_long_term_variability	0
mean_value_of_long_term_variability	0
histogram_width	0
histogram_min	0
histogram_max	0
histogram_number_of_peaks	0
histogram_number_of_zeroes	0
histogram_mode	0
histogram_mean	0
histogram_median	0
histogram_variance	0
histogram_tendency	0
fetal_health	0
dtype: int64	

STEP 5: Checking duplicate values

```
[16] # To check duplicated values
print(df.duplicated().value_counts())
```

False	1693
True	7
dtype: int64	

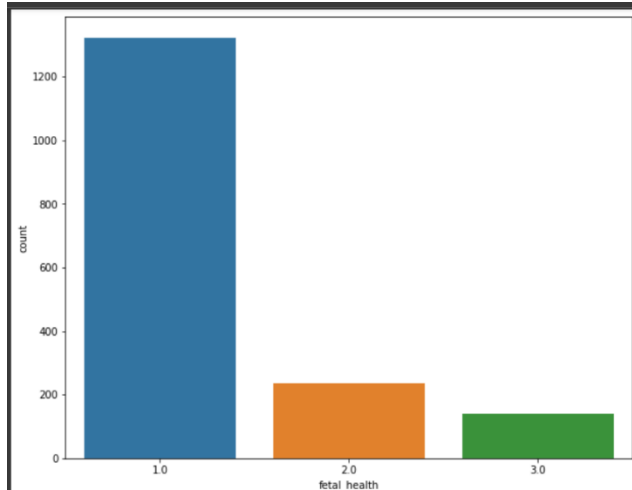
STEP 6: Checking class imbalance

```

✓ 0s # To check class imbalance

import seaborn as sns
plt.rcParams["figure.figsize"] = (10, 8)
df['fetal_health'].value_counts()
sns.countplot(x='fetal_health', data=df)
plt.show()

```



STEP 7: Dealing with class imbalance

```

[18] # To deal with class imbalance
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split
os = SMOTE(random_state=0)
X_class_train, X_test, y_class_train, y_test = train_test_split(X, Y, test_size=0.3, random_state=0)
columns = X_class_train.columns
data_X, data_y = os.fit_resample(X_class_train, y_class_train)
smoted_X = pd.DataFrame(data=data_X, columns=columns )
smoted_y= pd.DataFrame(data=data_y, columns=['fetal_health'])
X = smoted_X
Y = smoted_y

```

STEP 8: Making training data

```
[19] # splitting data into train test data
from sklearn.neural_network import MLPRegressor
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, random_state=1, test_size=0.2)

# To scale our data
sc_X = StandardScaler()
sc_X.fit(X_train)
X_train_scaled=sc_X.fit_transform(X_train)
X_test_scaled=sc_X.transform(X_test)

[21] # To get the columns name to make the scaled output as data frame,
columns_value_new=X_train.columns
test_X_Scaled_Except = pd.DataFrame(X_train_scaled, columns=columns_value_new)

[22] # Identify significant and independent features using correlation matrix without target variable
import seaborn as sns
plt.rcParams["figure.figsize"] = (24, 8)
sns.heatmap(test_X_Scaled_Except.corr(),annot=True);
```

STEP 9: Experimenting with different algorithms using scaled data

Here we get 73% accuracy using Naïve bayes, 96% accuracy using Decision tree and 97% accuracy using Random forest.

Here we can see the 97% accuracy using Random Forest algorithm.

```
✓ [34] from sklearn.ensemble import RandomForestClassifier
0.9 [34] clf_rf = RandomForestClassifier(max_depth=20, random_state=0).fit(X_train_scaled, y_train)
rf_pred = clf_rf.predict(X_test_scaled)

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: DataConversionWarning: A co

[35] print(accuracy_score(y_test, rf_pred))

0.969147005444646

# to see the accuracy of model
print(classification_report(y_test, rf_pred))
```

	precision	recall	f1-score	support
1.0	0.98	0.93	0.96	183
2.0	0.94	0.98	0.96	185
3.0	0.99	0.99	0.99	183
accuracy			0.97	551
macro avg	0.97	0.97	0.97	551
weighted avg	0.97	0.97	0.97	551
