MERISKILL

INTERNSHIP

APPLY HERE

TABLE OF CONTENTS

INTRODUCTION

A comprehensive report that encapsulates the essence of the three intriguing projects offered by MeriSKILL. These projects, centered within the realm of data analytics, are meticulously designed to unravel patterns and insights concealed within intricate datasets. The focus of these projects includes:

1. Sales Analysis: A deep dive into sales data, aimed at extracting valuable insights to enhance strategic decision-making.

2. Diabetic Patients: Exploring the intricacies of diabetic patient data to identify trends and patterns that can aid in improving healthcare outcomes.

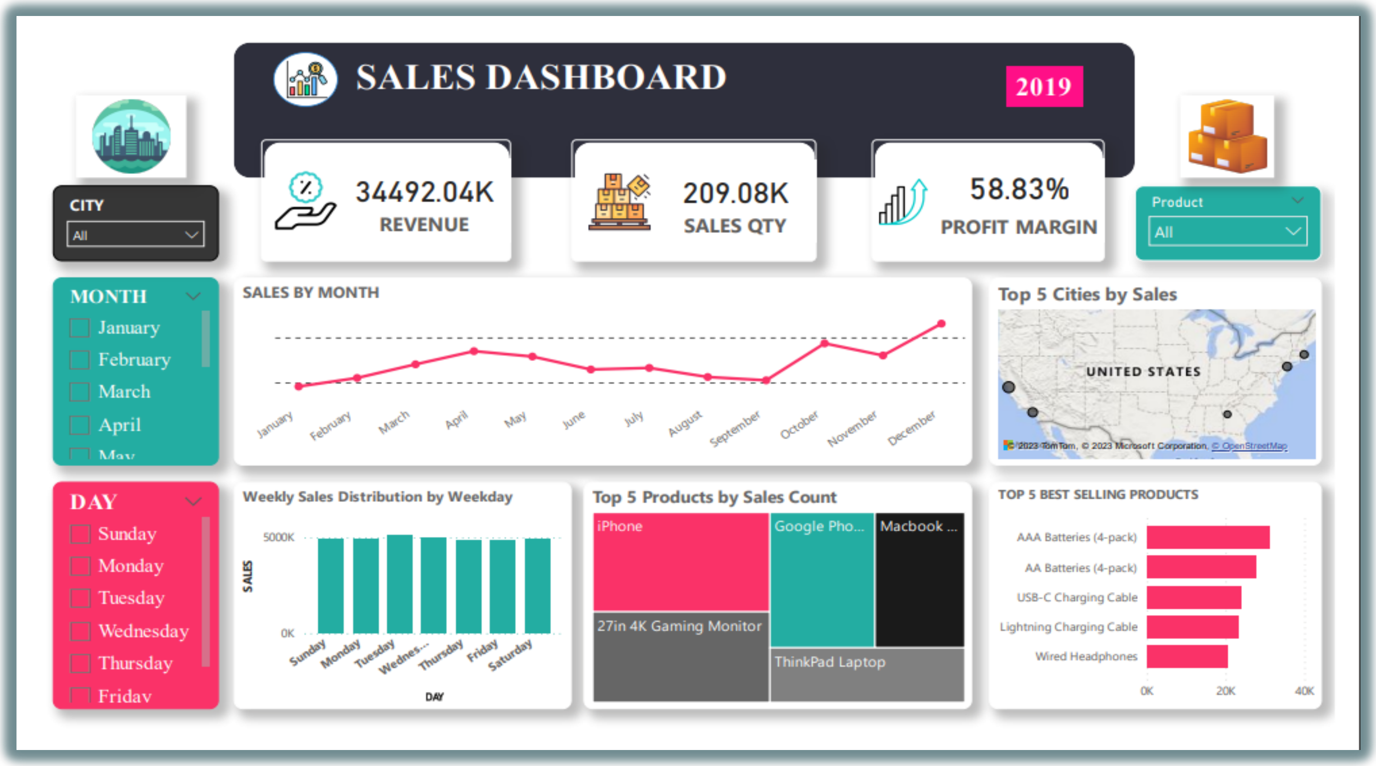
3. HR Analytics: Delving into the world of human resources, with a keen eye on data analysis to optimize talent management and organizational performance.

These projects, tailored for aspiring data analysts, offer a choice of analytical tools including Power BI, SQL, or Tableau to facilitate the extraction of critical insights. Through meticulous analysis and a keen understanding of the data, these projects seek to address pressing questions and uncover meaningful conclusions.

I have attached a Google Drive link with all the required codes and icons.

CLICK HERE

1. SALES DATA ANALYSIS



PURPOSE:

Analyze sales data to identify trends, top-selling products, and revenue metrics for business decision-making.

DESCRIPTION:

In this project, you will dive into a large sales dataset to extract valuable insights. You will explore sales trends over time, identify the best-selling products, calculate revenue metrics such as total sales and profit margins, and create visualizations to present your findings effectively. This project showcases your ability to manipulate and derive insights from large datasets, enabling you to make data-driven recommendations for optimizing sales strategies.

COLUMN DESCRIPTION FOR SALES DATA ANALYSIS:

- ORDER ID
- PRODUCT
- QUANTITY ORDERED
- PRICE EACH
- ORDER DATE
- PURCHASE ADDRESS
- MONTH
- SALES
- CITY
- HOUR

EXECUTION PART

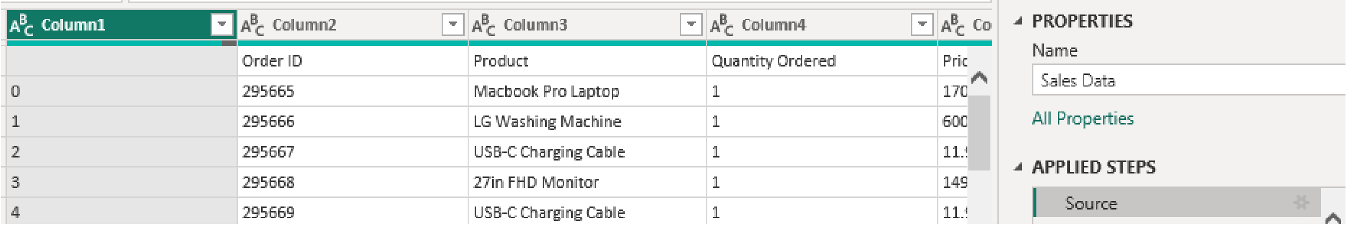
TRANSFORMATION OF DATA

STEP 1:

I downloaded the dataset, uploaded it using the 'Get Data' option in Power BI, and then proceeded to transform the data.

STEP 2:

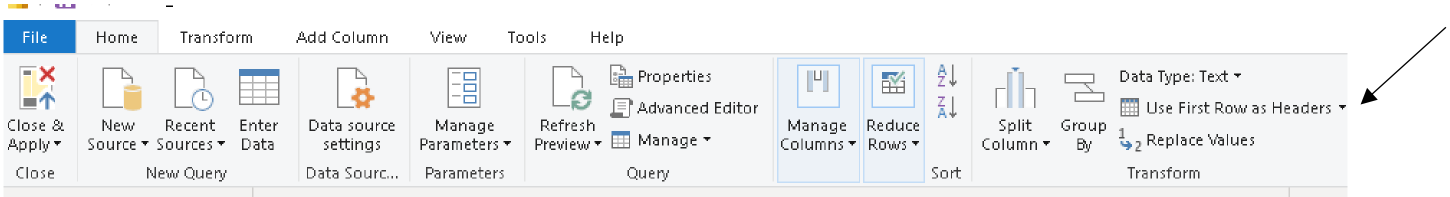
The column headers are identified in the first row and should be kept as headers by following the steps below.

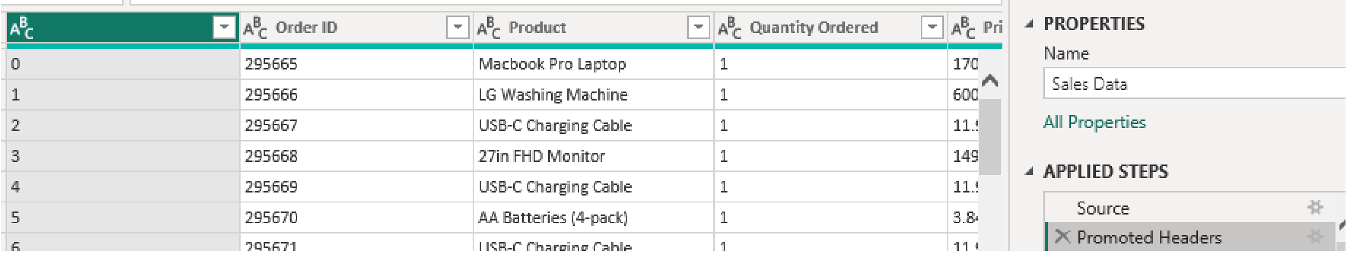


Column1	Column2	Column3	Column4	Co
	Order ID	Product	Quantity Ordered	Pric
0	295665	Macbook Pro Laptop	1	170
1	295666	LG Washing Machine	1	600
2	295667	USB-C Charging Cable	1	11.
3	295668	27in FHD Monitor	1	149
4	295669	USB-C Charging Cable	1	11.

PROPERTIES
Name
Sales Data
All Properties

APPLIED STEPS
Source





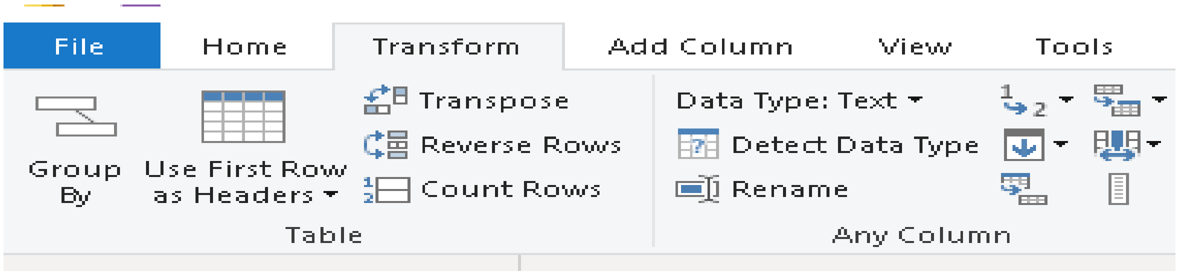
	Order ID	Product	Quantity Ordered	Pri
0	295665	Macbook Pro Laptop	1	170
1	295666	LG Washing Machine	1	600
2	295667	USB-C Charging Cable	1	11.
3	295668	27in FHD Monitor	1	149
4	295669	USB-C Charging Cable	1	11.
5	295670	AA Batteries (4-pack)	1	3.8
6	295671	USB-C Charging Cable	1	11.

PROPERTIES
Name
Sales Data
All Properties

APPLIED STEPS
Source
Promoted Headers

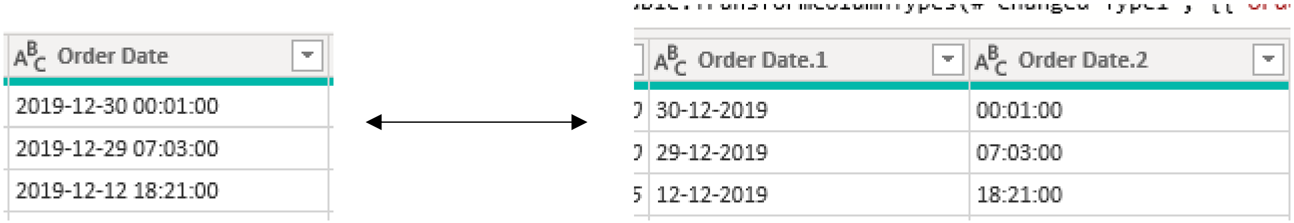
STEP 3:

- After promoting the headers, navigate to the 'Transform' tab and select 'Detect Data Type.' This action will automatically identify the data type of each column and convert them as needed**.**



STEP 4:

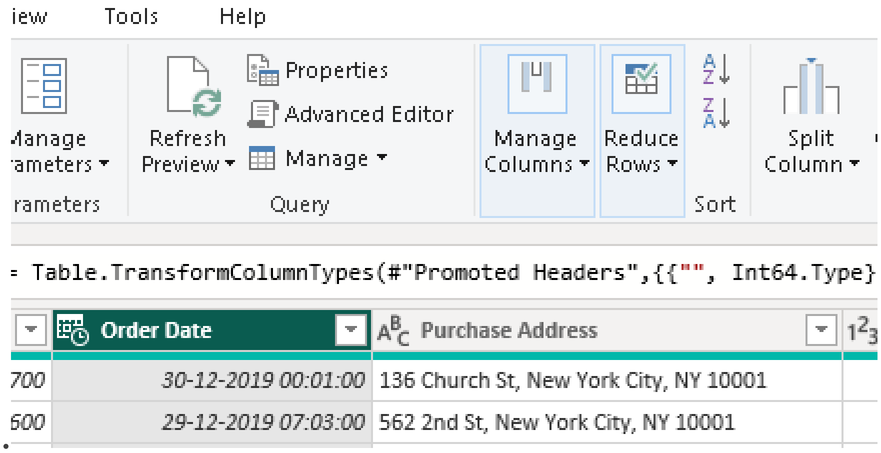
- Split the datetime into date and time stamp



Order Date
2019-12-30 00:01:00
2019-12-29 07:03:00
2019-12-12 18:21:00

Order Date.1	Order Date.2
30-12-2019	00:01:00
29-12-2019	07:03:00
12-12-2019	18:21:00

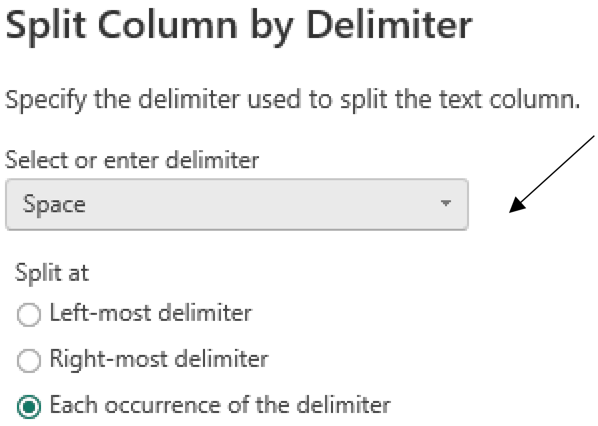
- The aforementioned process starts with selecting the desired column. Following the selection, the option to split the column becomes visible.



```
= Table.TransformColumnTypes(#"Promoted Headers",{{"", Int64.Type}
```

	Order Date	Purchase Address
700	30-12-2019 00:01:00	136 Church St, New York City, NY 10001
600	29-12-2019 07:03:00	562 2nd St, New York City, NY 10001

- Choose the 'Split Column' option and select the space as the delimiter.
- Upon completing the data transformation, click on 'Close & Apply' located at the top left.
- Remember, this step is crucial after any data transformation process.



Split Column by Delimiter

Specify the delimiter used to split the text column.

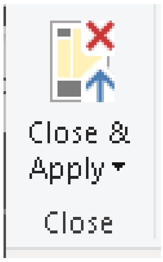
Select or enter delimiter: Space

Split at

○ Left-most delimiter

○ Right-most delimiter

● Each occurrence of the delimiter

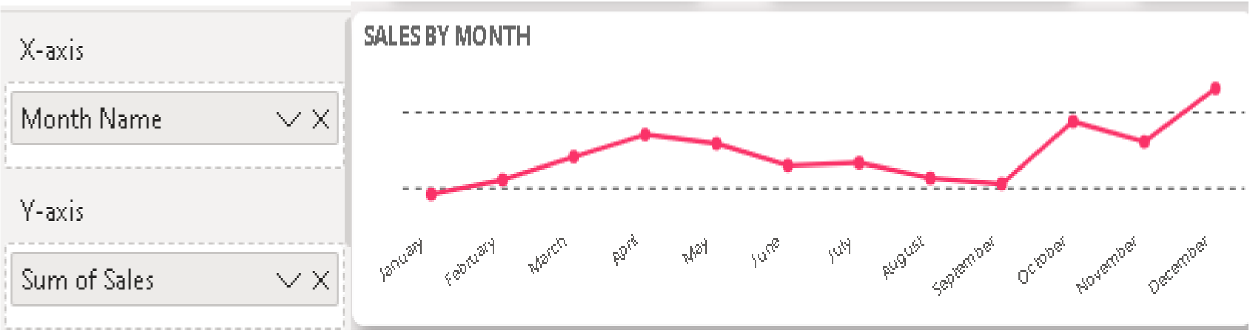


VISUALIZATION OF DATA

STEP 1:

Sales trend over time using the line chart

- Simply click on the Month name and Sales column, drag it to the desired position.

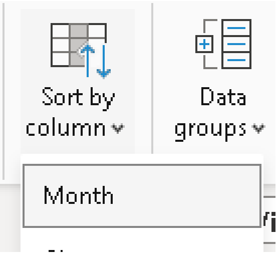


X-axis
Month Name
Y-axis
Sum of Sales

- To create a Chronological order for the months, follow these steps:

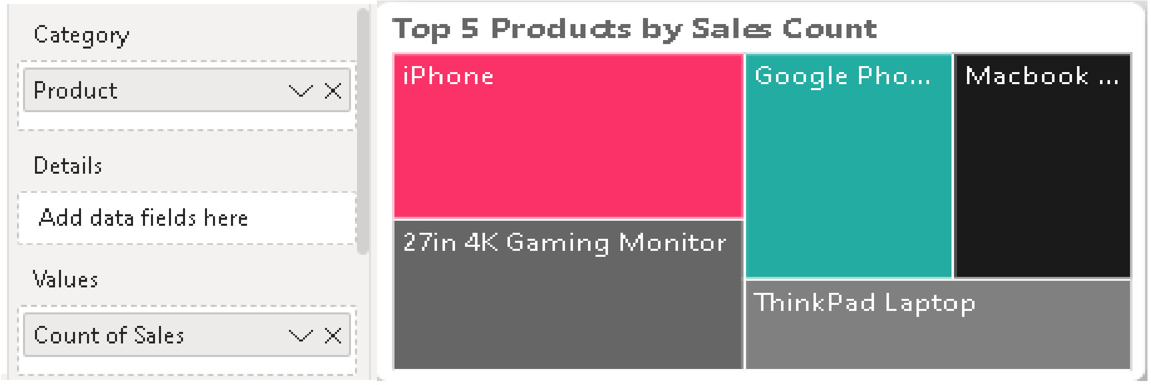
1. Select the column containing the months.
2. Navigate to the "Column Tools" and choose "Sort Column."
3. Select "Sort by Month Number" to sort the months in chronological order.

☑ Month Name ···



STEP 2: Best selling products using tree map

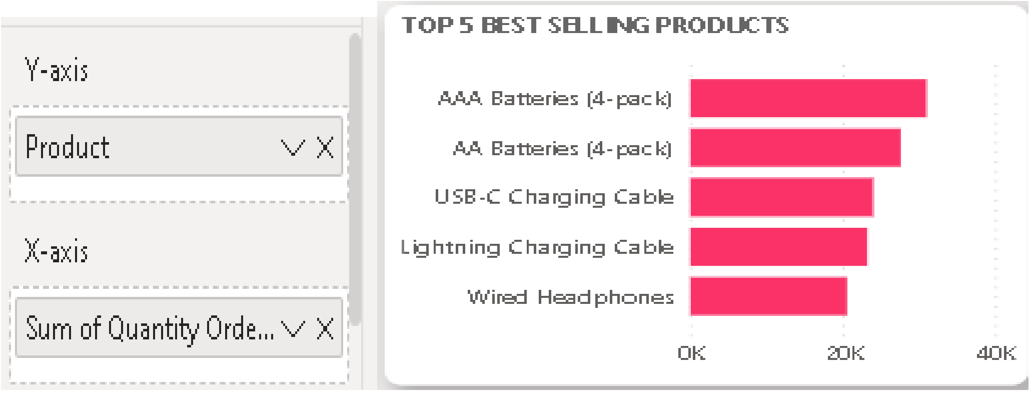
- To edit them for background color and font size, access the "Format" option for the visualization and adjust the settings as desired.



Category
Product
Details
Add data fields here
Values
Count of Sales

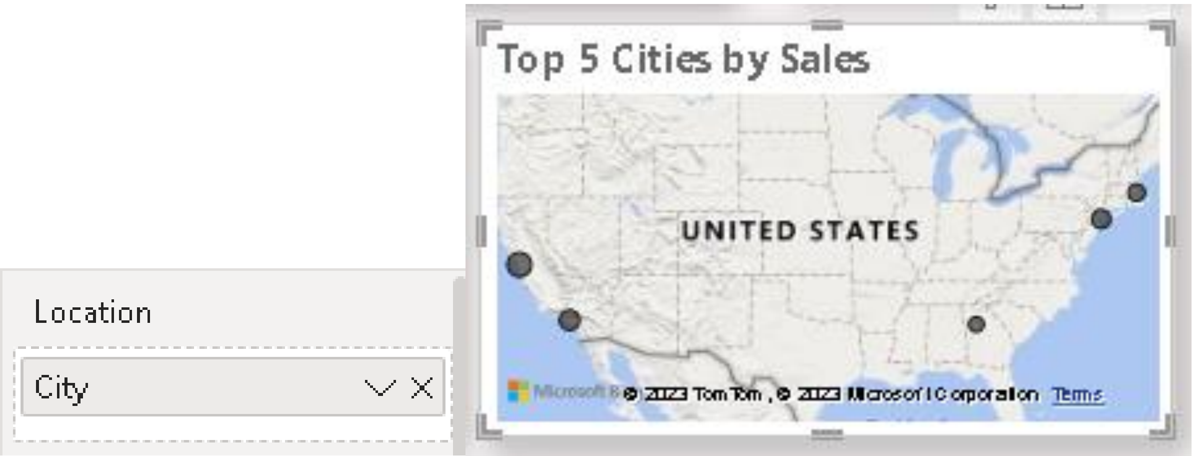
STEP 3: Top 5 best selling product using stacked bar chart

1. To manipulate the visualization, perform the following steps:
   - Drag and drop the "Product" into the Y-axis.
   - Place the "Quantity" into the X-axis for appropriate ordering.



Y-axis
Product
X-axis
Sum of Quantity Orde...

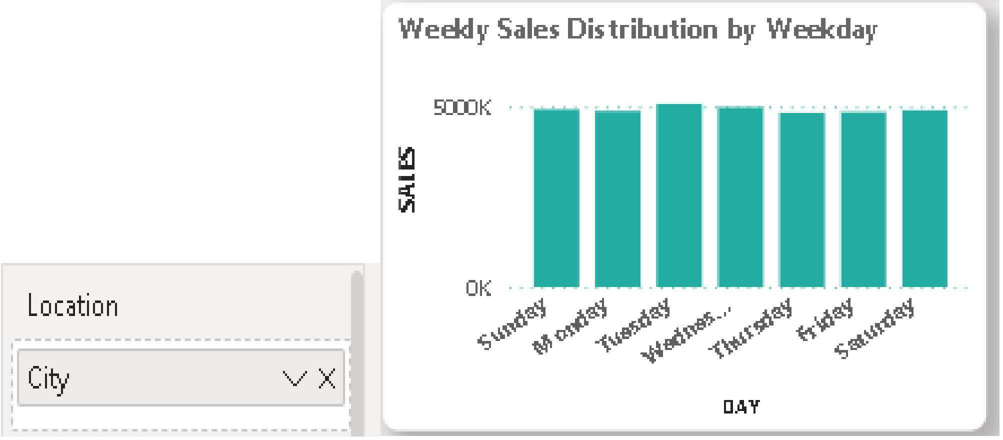
STEP 4: Top 5 cities by sales using map


Top 5 Cities by Sales
UNITED STATES
Microsoft Bing © 2023 TomTom, © 2023 Microsoft Corporation Terms

Location

City

STEP 5: Weekly sales distribution by weekday using column chart

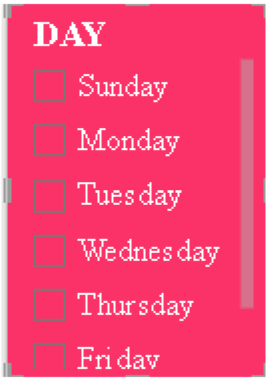


Location

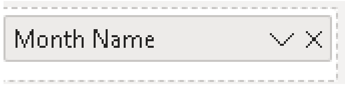
City

STEP 6: Slicer is used to make this kind of visual

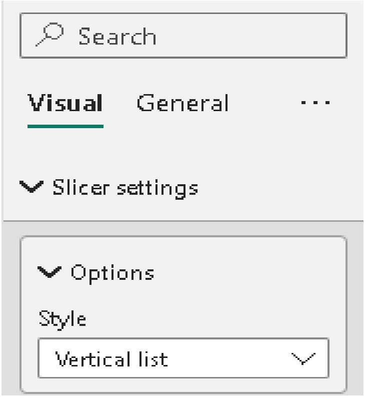
- To create a slicer visualization, drag and drop the "Month Name" field into the slicer option. To display the slicer in a vertical list, access the slicer settings and choose the option for a vertical column layout.



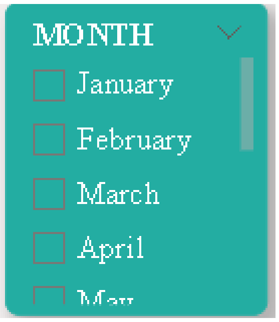
DAY
□ Sunday
□ Monday
□ Tuesday
□ Wednesday
□ Thursday
□ Friday



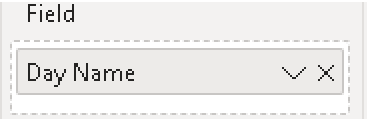
Month Name



Style: Vertical list

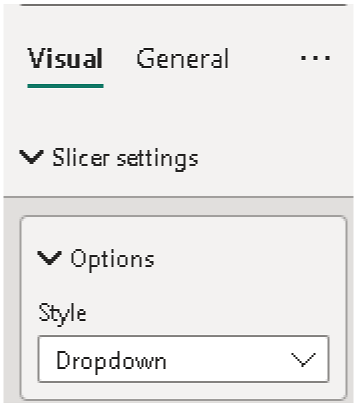


MONTH
□ January
□ February
□ March
□ April
□ May

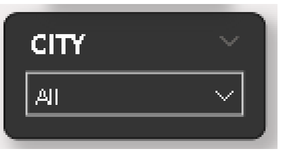


Field: Day Name

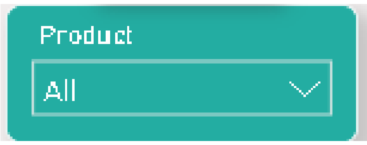
The example below showcases the slicer functionality. The slicer setting has been adjusted to display as a dropdown menu.



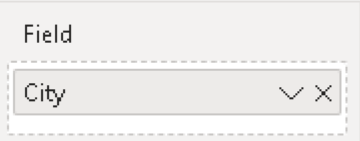
Style: Dropdown



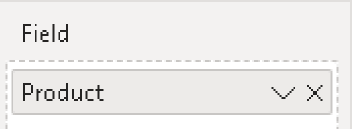
CITY
All



Product
All



Field: City



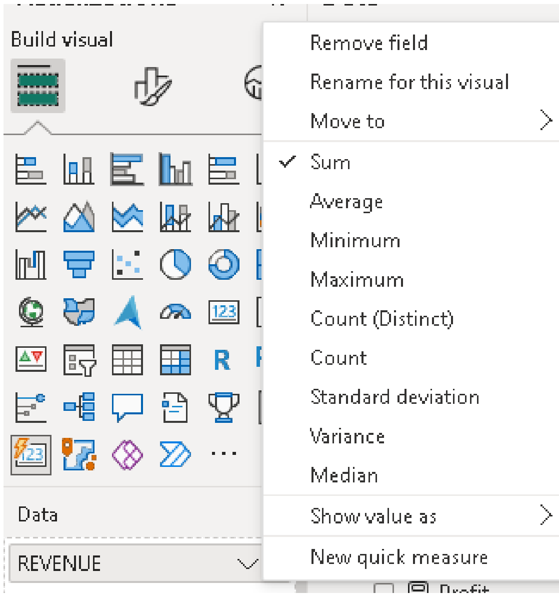
Field: Product

STEP 7: To find the revenue metrics:

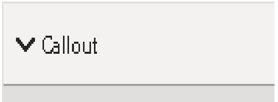
- **Total profit:** Sum up the net profit from all sales transactions.
- **Sales quantity:** Calculate the total number of units sold.
- **Profit margin:** Compute the ratio of net profit to total revenue, usually expressed as a percentage**.**

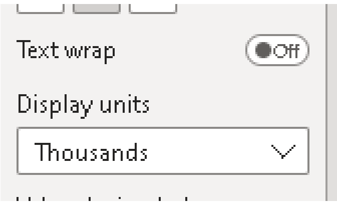
REVENUE = SUM OF SALES

1. Select the "Card" visualization type, then drag and drop the "Sales" into the designated field. Convert it to the "SUM" aggregation.
2. Additionally, adjust the display units to show values in millions, billions, trillions, or hundreds, and customize the number of decimal places as needed.

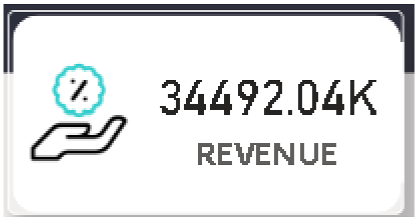


THE CARD VISUAL CAN BE DESIGNED USING THE "FORMAT VISUAL"




Text wrap ● Off

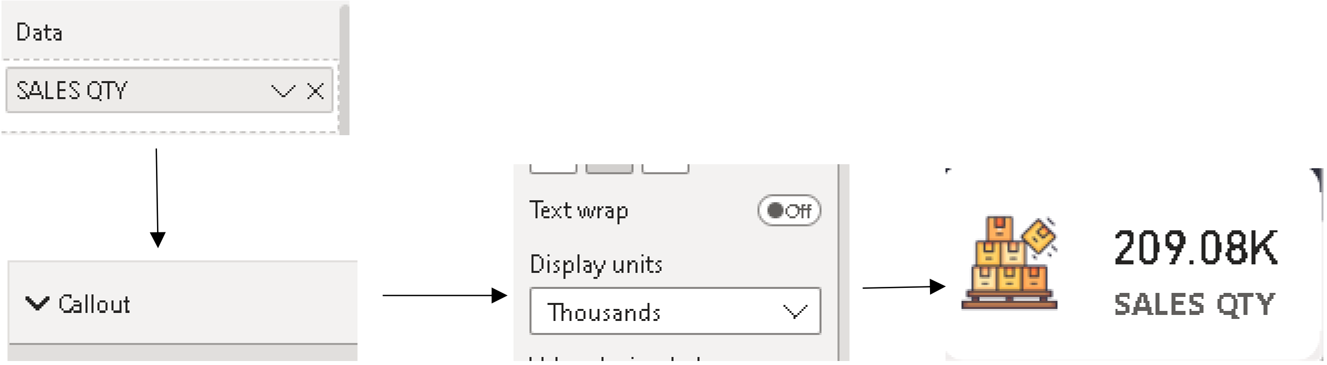
Display units: Thousands


34492.04K

REVENUE

Sales quantity

- Select the "Card" visual, then drag and drop the "Quantity Ordered" int the designated field.
- Access the "Format" option for the visual, and adjust the callout value to change the display unit of the quantity ordered as desired.



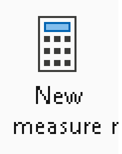
Text wrap Off

Display units Thousands

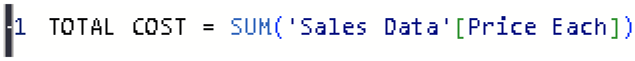
Profit margin

PROFIT MARGIN = ((TOTAL SALES -TOTAL COST)/TOTAL SALES)*100;

- Click on new measure

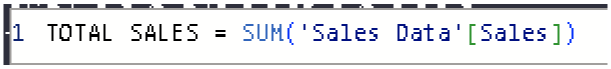


- Find the total cost by using the new measure



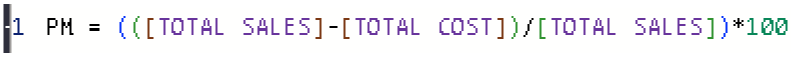
```
TOTAL COST = SUM('Sales Data'[Price Each])
```

- Find the total sales by using the new measure



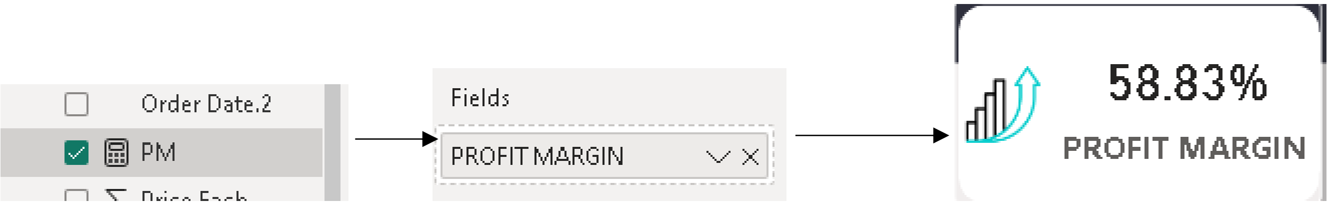
```
TOTAL SALES = SUM('Sales Data'[Sales])
```

- Find the profit margin by using this formula in the measure.



```
PM = (([TOTAL SALES]-[TOTAL COST])/[TOTAL SALES])*100
```

- Choose the measure created and place it in the card visual and design using the “Format visual”



□ Order Date.2

☑ PM

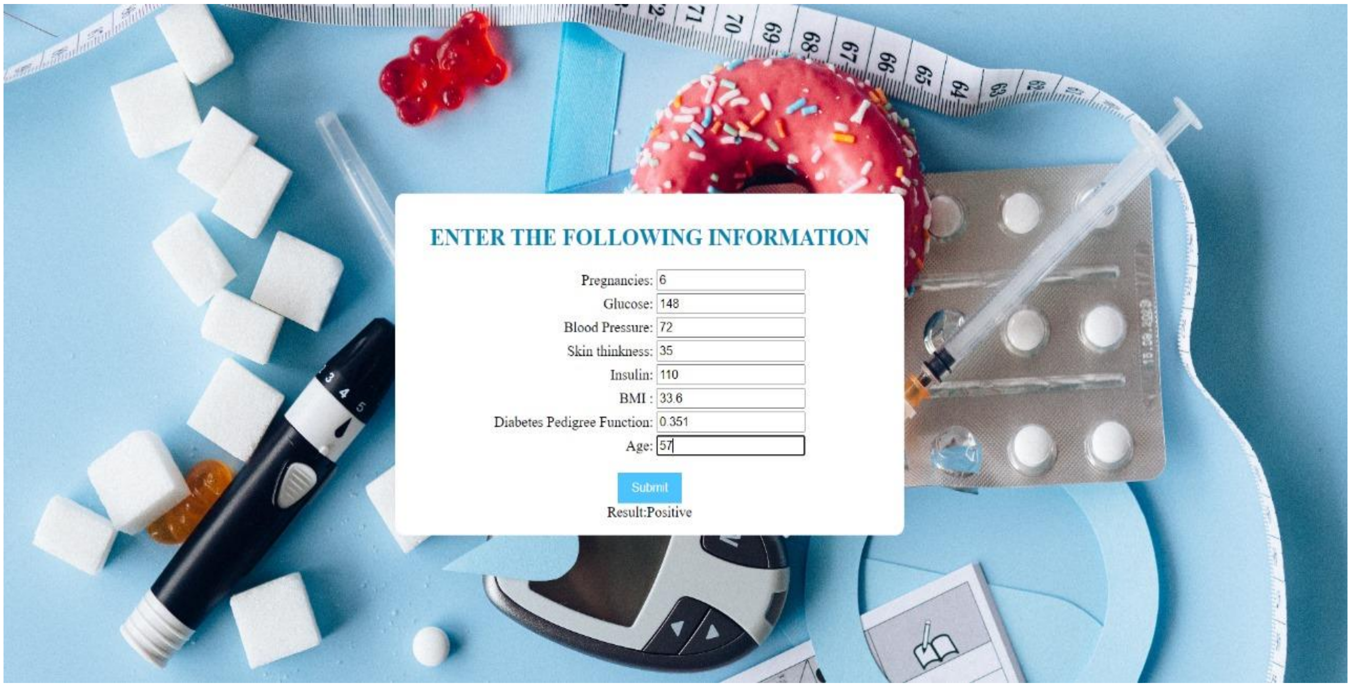
Fields

PROFIT MARGIN

58.83%

PROFIT MARGIN

2.DIABETES PATIENTS



ENTER THE FOLLOWING INFORMATION

Pregnancies: 6

Glucose: 148

Blood Pressure: 72

Skin thinkness: 35

Insulin: 110

BMI : 33.6

Diabetes Pedigree Function: 0.351

Age: 57

Result:Positive

ABOUT DATASET

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether a patient has diabetes based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, **all patients here are females at least 21 years old** of Pima Indian heritage.

COLUMN DESCRIPTION FOR DIABETES DATA:

- Pregnancies
- Glucose
- Blood Pressure
- Skin Thickness
- Insulin
- BMI
- Diabetes
- Age
- Outcome

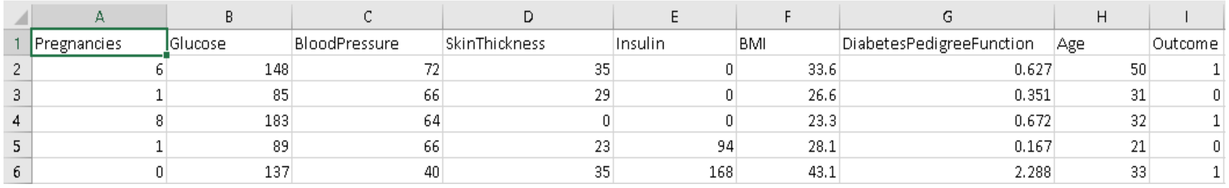
From the data set in the (.csv) File We can find several variables, some of them are independent (several medical predictor variables) and only one target dependent variable (Outcome).

STEP 1:

- I downloaded and opened my dataset and attempted to understand the type of analysis expected. Upon examination, I discovered that the dataset contains data suitable for predictive analysis.

- Predictive analysis is a type of analysis where this means understanding the probable future trends and behavior

- To derive predictions from the given dataset, I partitioned the data into independent and dependent variables. Upon analysis, I observed that all seven variables i.e. Pregnancies, Glucose, Blood pressure, skin thickness, Insulin, BMI , Diabetes, Age are independent, while 'Outcome' serves as the dependent variable.
- From the given dataset, 0 represents that the person is not diabetic and 1 shows that the person is diabetic.



	A	B	C	D	E	F	G	H	I
1	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
2	6	148	72	35	0	33.6	0.627	50	1
3	1	85	66	29	0	26.6	0.351	31	0
4	8	183	64	0	0	23.3	0.672	32	1
5	1	89	66	23	94	28.1	0.167	21	0
6	0	137	40	35	168	43.1	2.288	33	1

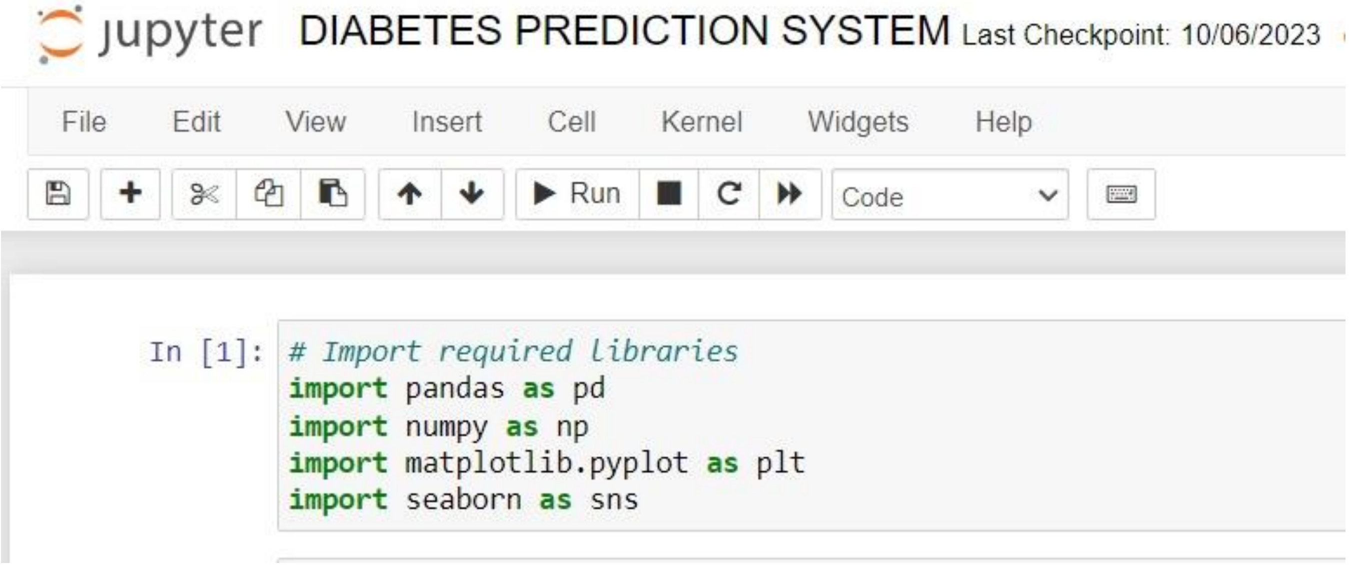
STEP 2:

1. Install **Jupyter notebook** – **click here**
2. Import the important packages
3. Before importing , some packages must be installed in jupyter notebook

```
pip install numpy
pip install pandas
pip install scikit-learn
pip install matplotlib
pip install seaborn
```

STEP 3:

a) IMPORTING REQUIRED LIBRARIES



```
In [1]: # Import required libraries
        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
```

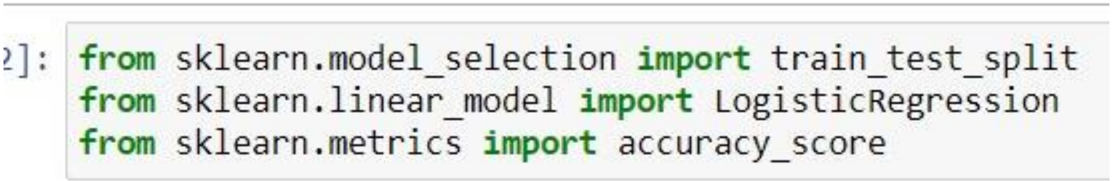
Pandas: Data analysis and manipulation library for working with structured data using Data Frame and Series.

NumPy: Numerical computing library supporting large, multi-dimensional arrays and matrices, with high-level mathematical functions.

Seaborn: Statistical data visualization library for creating attractive and informative graphics, based on Matplotlib.

Matplotlib: Comprehensive plotting library providing interface for creating various plots like line, scatter, bar, and histograms.

b) IMPORTING IMPORTANT LIBRARIES FOR PREDICTION



```
2]: from sklearn.model_selection import train_test_split
    from sklearn.linear_model import LogisticRegression
    from sklearn.metrics import accuracy_score
```

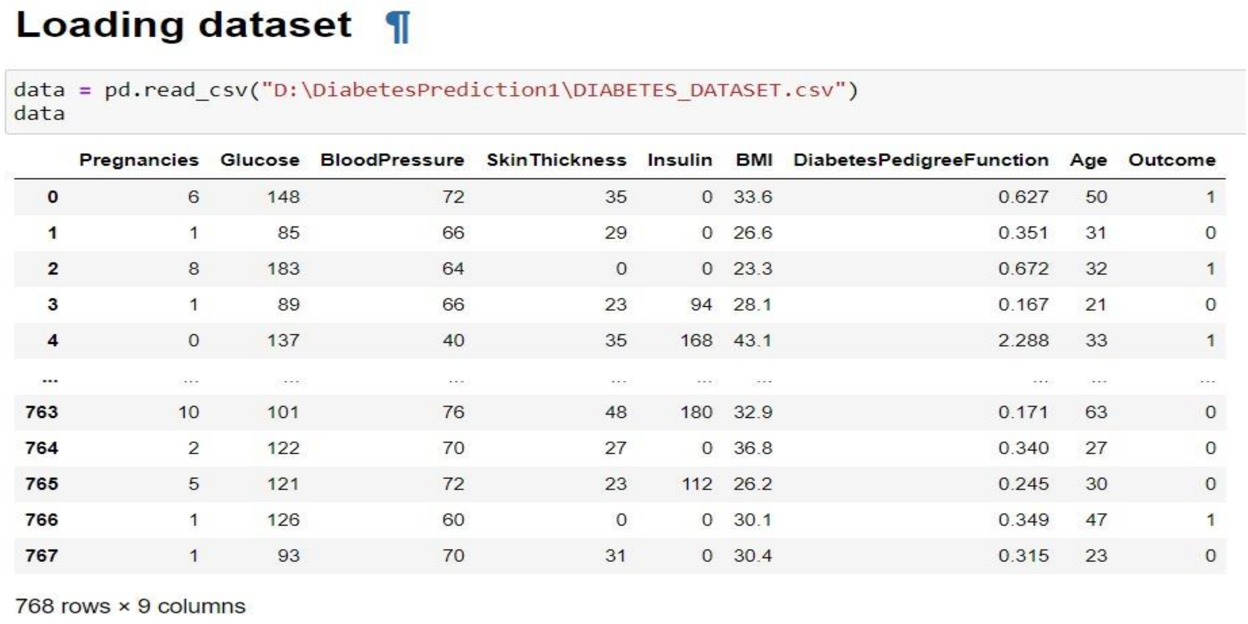
Train Test Split: Technique for splitting data into training and testing sets to assess model performance.

Logistic Regression: Method for predicting the probability of a binary outcome using the logistic function.

Accuracy: Metric measuring the proportion of correctly classified instances in a classification model.

Sklearn: Python's Scikit-learn, a powerful machine learning library providing tools for data analysis and model building.

c) LOADING THE DATASET



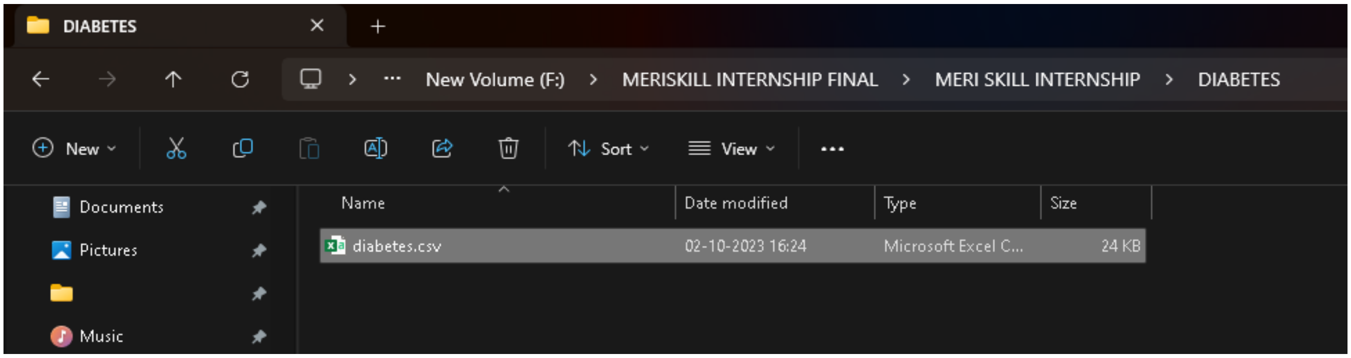
Loading dataset ¶

```
data = pd.read_csv("D:\DiabetesPrediction1\DIABETES_DATASET.csv")
data
```

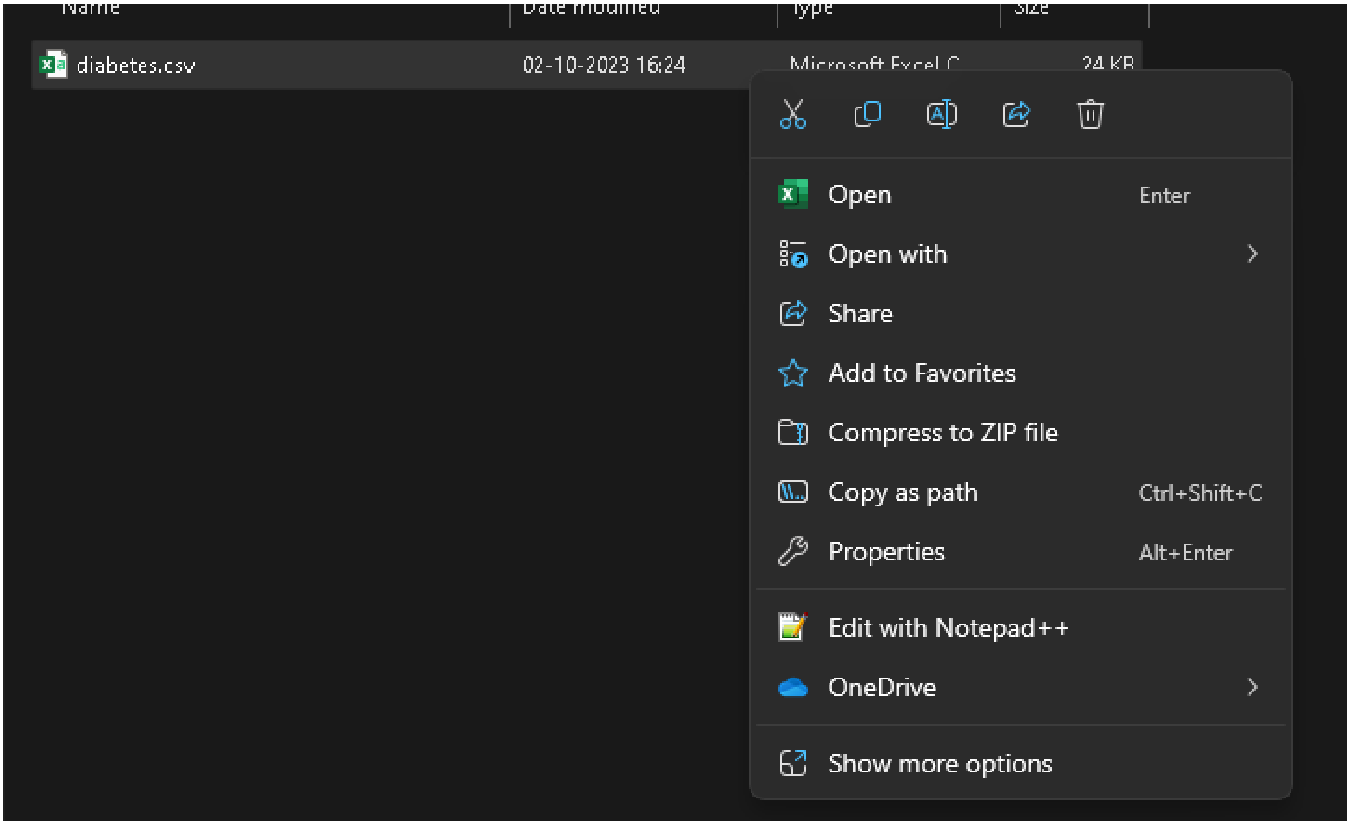
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

- To get the path of the file

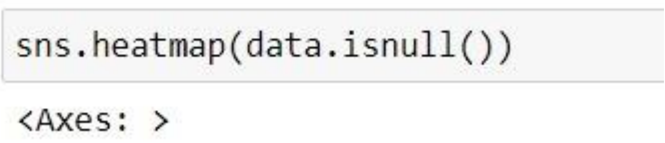

DIABETES
New Volume (F:) > MERISKILL INTERNSHIP FINAL > MERI SKILL INTERNSHIP > DIABETES
New
Sort
View
Documents
Pictures
Music

Name	Date modified	Type	Size
diabetes.csv	02-10-2023 16:24	Microsoft Excel C...	24 KB



- Click “copy as path” and paste it at the “data = pd.read_csv(“**paste here**”)”

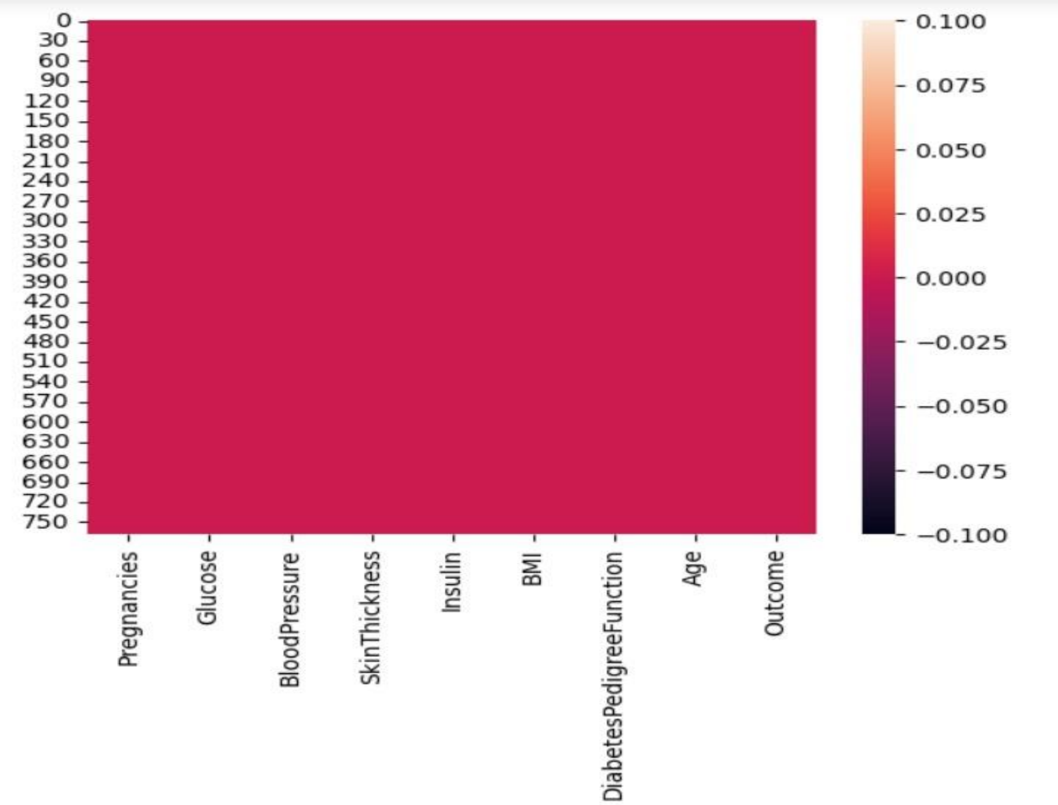
d)CHECKING FOR MISSING VALUES



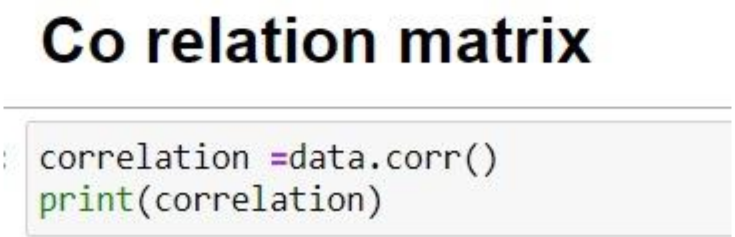
```
sns.heatmap(data.isnull())
```

```
<Axes: >
```

O/P

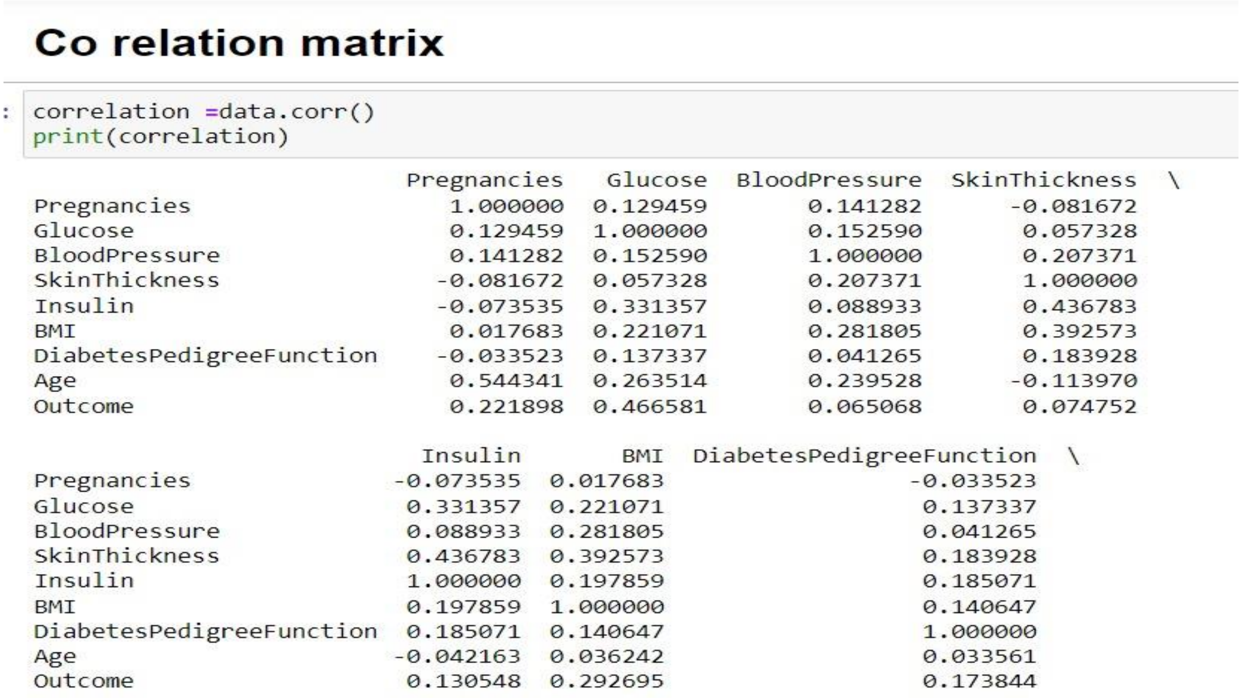


e) CO RELATION MATRIX



Co relation matrix

```
: correlation =data.corr()
  print(correlation)
```

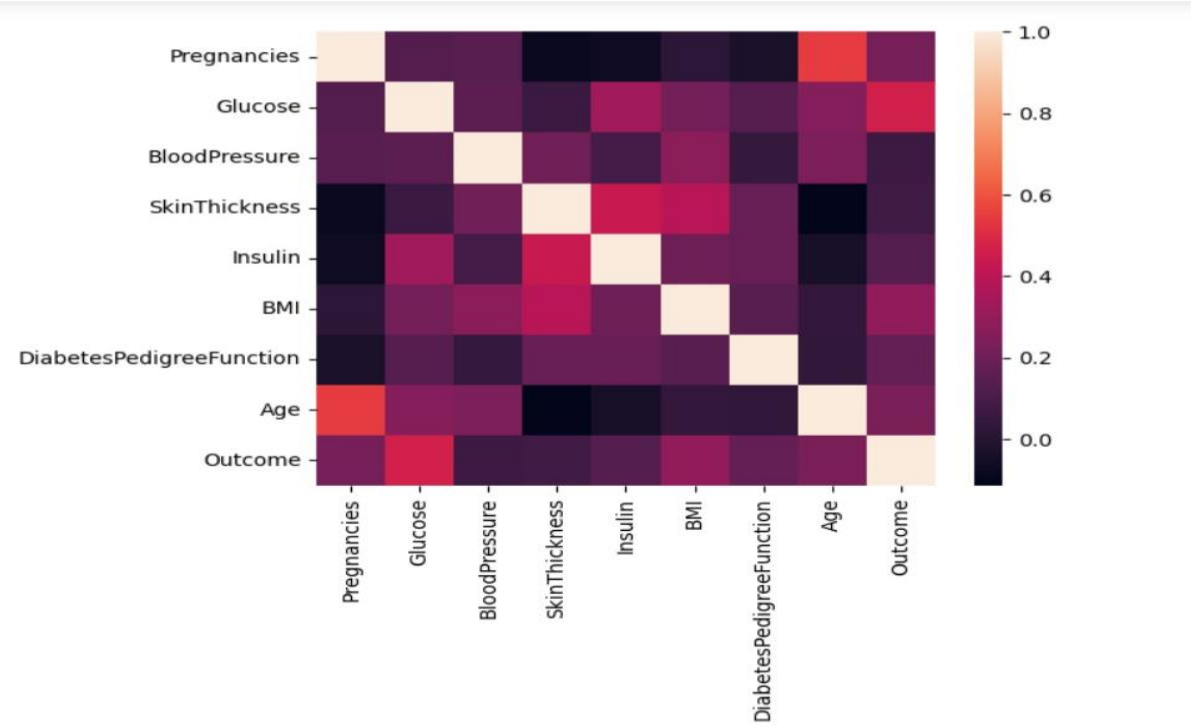


Co relation matrix

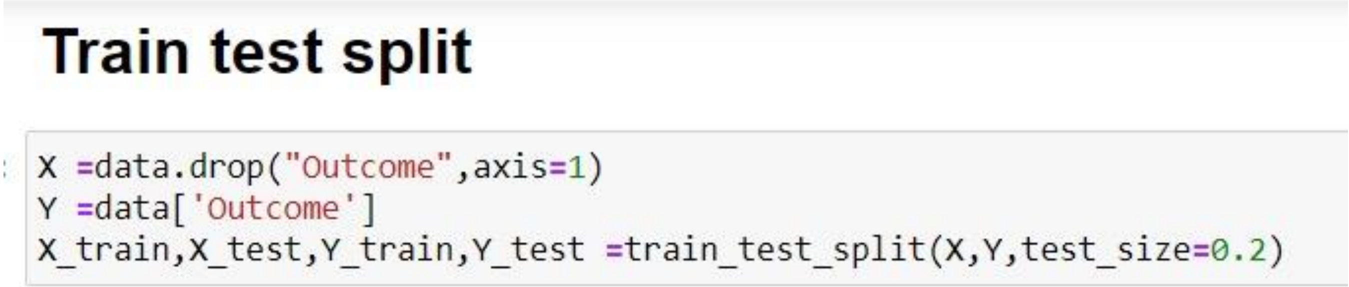
```
: correlation =data.corr()
  print(correlation)
```

```
                          Pregnancies   Glucose  BloodPressure  SkinThickness  \
Pregnancies                  1.000000  0.129459       0.141282      -0.081672
Glucose                      0.129459  1.000000       0.152590       0.057328
BloodPressure                0.141282  0.152590       1.000000       0.207371
SkinThickness               -0.081672  0.057328       0.207371       1.000000
Insulin                     -0.073535  0.331357       0.088933       0.436783
BMI                          0.017683  0.221071       0.281805       0.392573
DiabetesPedigreeFunction    -0.033523  0.137337       0.041265       0.183928
Age                          0.544341  0.263514       0.239528      -0.113970
Outcome                      0.221898  0.466581       0.065068       0.074752

                           Insulin       BMI  DiabetesPedigreeFunction  \
Pregnancies              -0.073535  0.017683                 -0.033523
Glucose                   0.331357  0.221071                  0.137337
BloodPressure             0.088933  0.281805                  0.041265
SkinThickness             0.436783  0.392573                  0.183928
Insulin                   1.000000  0.197859                  0.185071
BMI                       0.197859  1.000000                  0.140647
DiabetesPedigreeFunction  0.185071  0.140647                  1.000000
Age                      -0.042163  0.036242                  0.033561
Outcome                   0.130548  0.292695                  0.173844
```

- **Visualizing the correlation**



f) TRAINING THE MODEL WITH THE HELP OF TRAIN TEST SPLIT



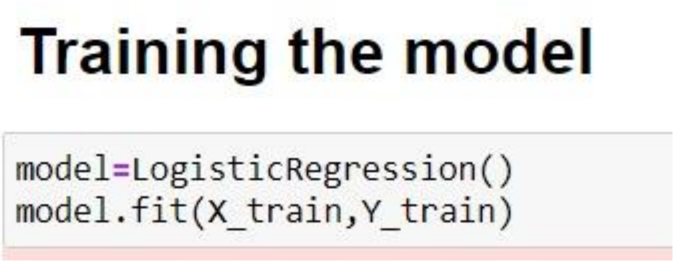
Train test split

```
X =data.drop("Outcome",axis=1)
Y =data['Outcome']
X_train,X_test,Y_train,Y_test =train_test_split(X,Y,test_size=0.2)
```

In X all the independent variables are stored

In Y the predictor variable(“OUTCOME”) is stored.

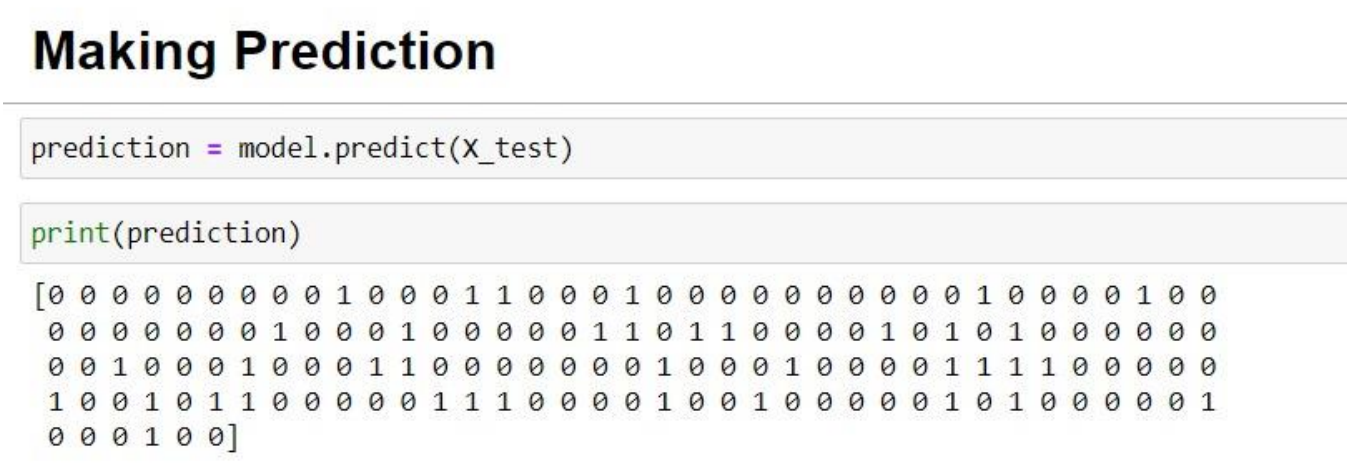
Train-test split is a technique used in machine learning to assess model performance. It divides the dataset into a training set and a testing set, with a 0.2 test size indicating that 20% of the data is used for testing and 80% for training.



Training the model

```
model=LogisticRegression()
model.fit(X_train,Y_train)
```

Fitting the X train and y train data into the variable called model.



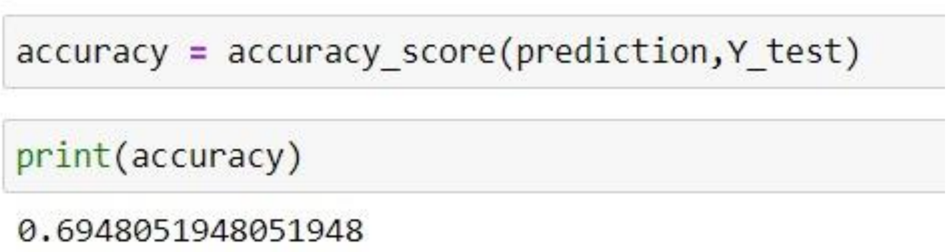
Making Prediction

```
prediction = model.predict(X_test)
```

```
print(prediction)
```

```
[0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0
 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 1 0 1 1 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0
 0 0 1 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 1 1 1 0 0 0 0 0
 1 0 0 1 0 1 1 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 1
 0 0 0 1 0 0]
```

After training the model, predictions are made using the test data, which comprises 20% of the total dataset.



```
accuracy = accuracy_score(prediction,Y_test)
```

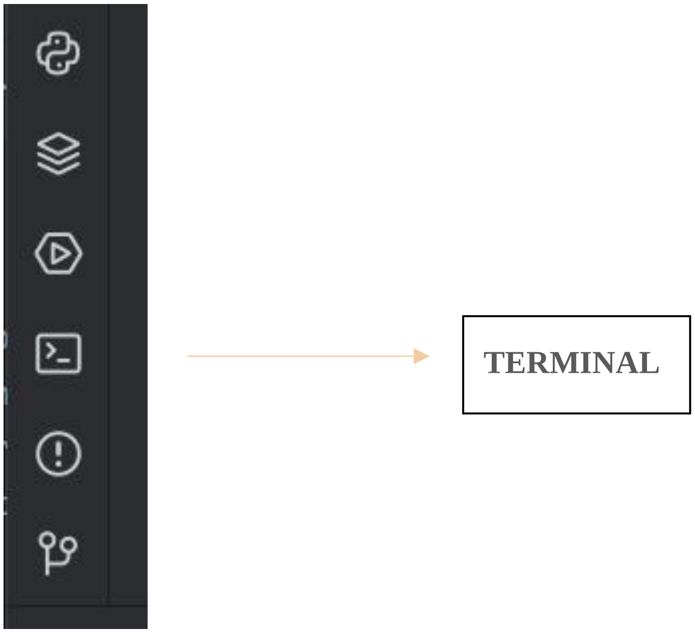
```
print(accuracy)
```

```
0.6948051948051948
```

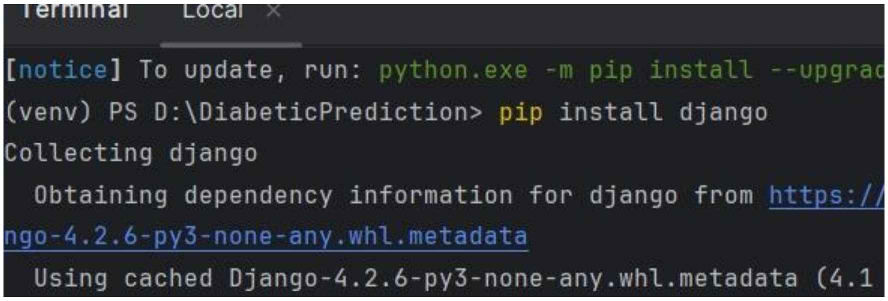
The accuracy of the model is then calculated and determined.

STEP 4:

- Install Pycharm community edition – click here
- Go to command prompt “conda install -c anaconda django”
- Open pycharm → New project →“Diabetes prediction” (folder name)
- Set Interpreter → Click here for explanation video
- Go to terminal

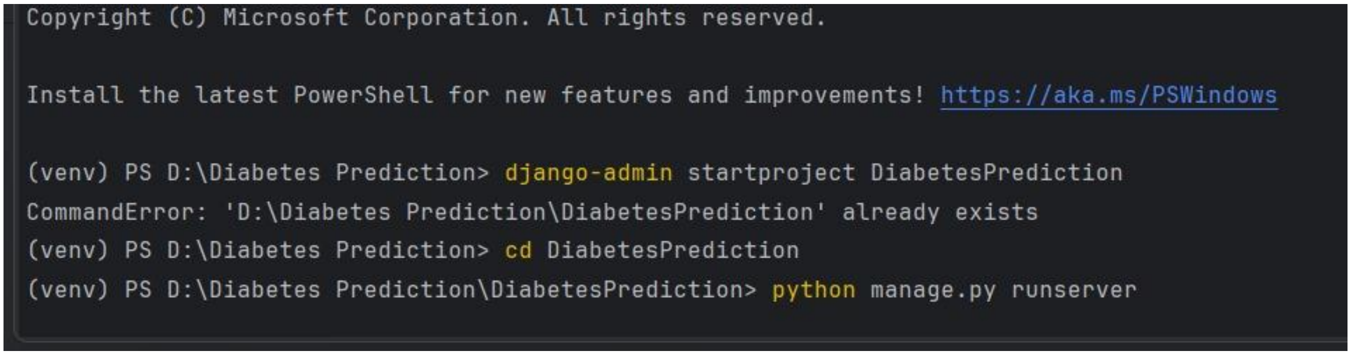


1. First command - pip install django

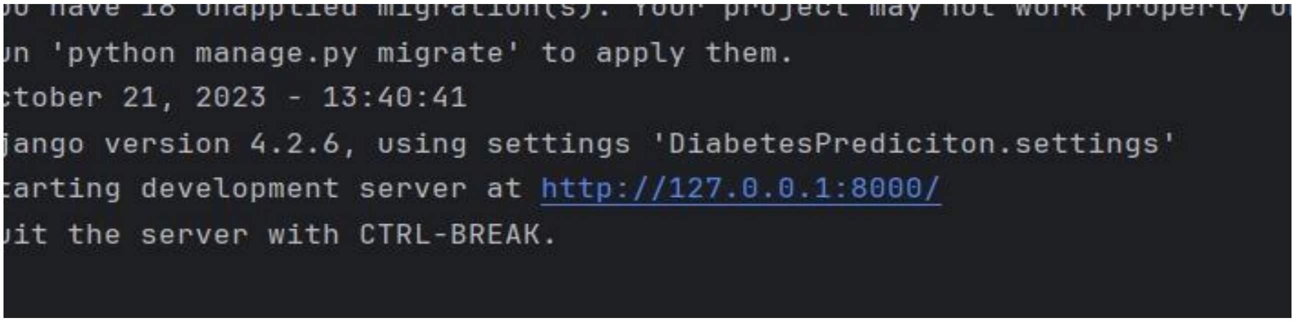


```
[notice] To update, run: python.exe -m pip install --upgrac
(venv) PS D:\DiabeticPrediction> pip install django
Collecting django
  Obtaining dependency information for django from https://
ngo-4.2.6-py3-none-any.whl.metadata
  Using cached Django-4.2.6-py3-none-any.whl.metadata (4.1
```

2.django-admin startproject “Give ur folder name without quotes”
3.cd “foldername”
4.python manage.py runserver
5.A link will be generated

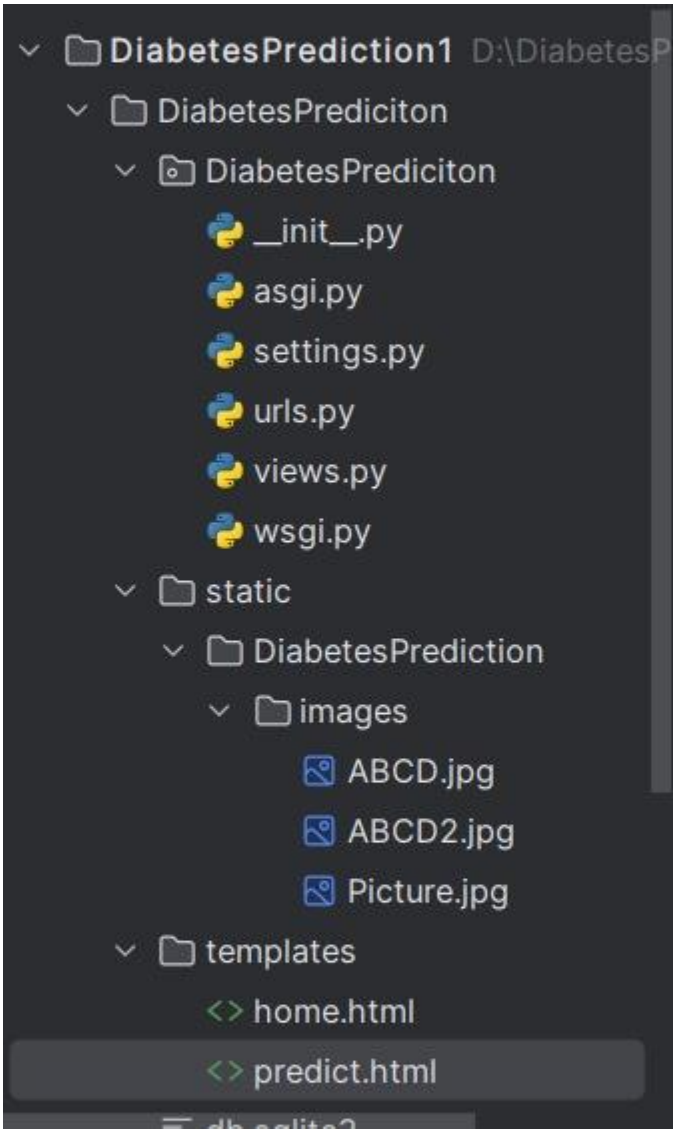


```
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

(venv) PS D:\Diabetes Prediction> django-admin startproject DiabetesPrediction
CommandError: 'D:\Diabetes Prediction\DiabetesPrediction' already exists
(venv) PS D:\Diabetes Prediction> cd DiabetesPrediction
(venv) PS D:\Diabetes Prediction\DiabetesPrediction> python manage.py runserver
```



```
ou have 18 unapplied migration(s). Your project may not work properly u
un 'python manage.py migrate' to apply them.
ctober 21, 2023 - 13:40:41
jango version 4.2.6, using settings 'DiabetesPrediciton.settings'
tarting development server at http://127.0.0.1:8000/
uit the server with CTRL-BREAK.
```

STEP 5:



```
DiabetesPrediction1  D:\DiabetesP
  DiabetesPrediciton
    DiabetesPrediciton
      __init__.py
      asgi.py
      settings.py
      urls.py
      views.py
      wsgi.py
    static
      DiabetesPrediction
        images
          ABCD.jpg
          ABCD2.jpg
          Picture.jpg
    templates
      home.html
      predict.html
    db.sqlite3
```

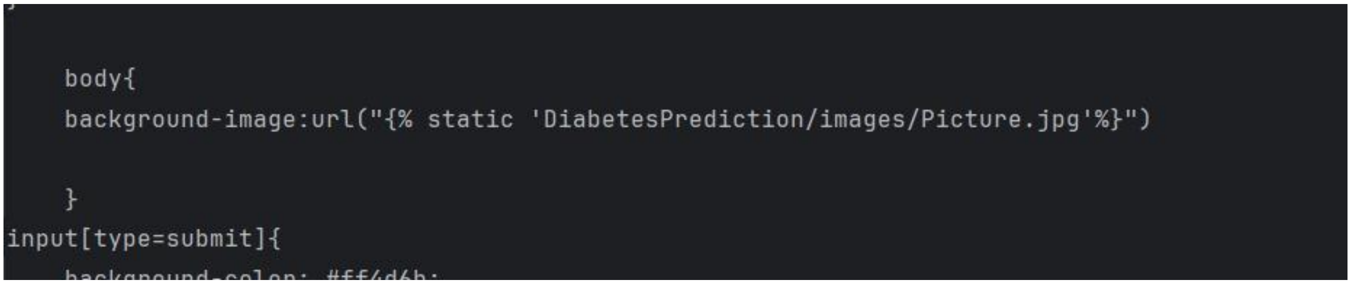
Right click and create a new directory "static" and "template"

Right click and create a new directory "Diabetes prediction" and another directory "images " to store the images given in google drive. Download, drag and Drop the images

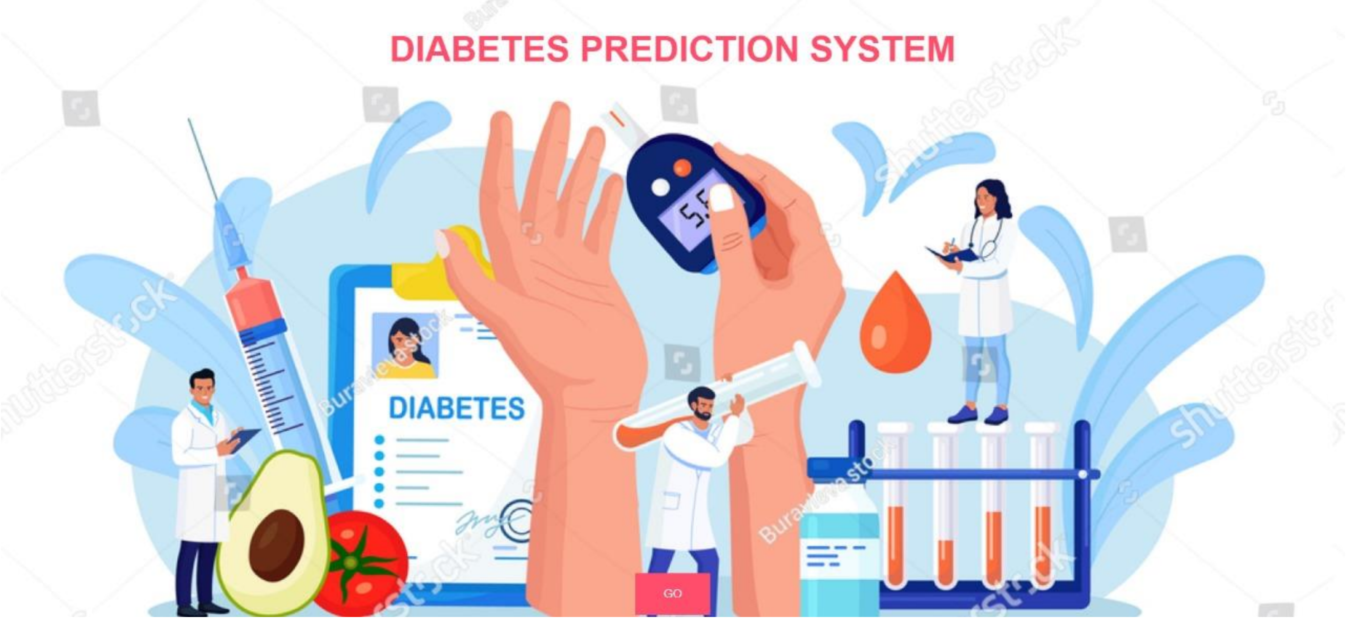
Inside template create two html files and copy paste the code from the google drive given

home.html

- This file consist of the design of the front page
- By right clicking on the image →copy path/reference, you will get the path and can paste it here.

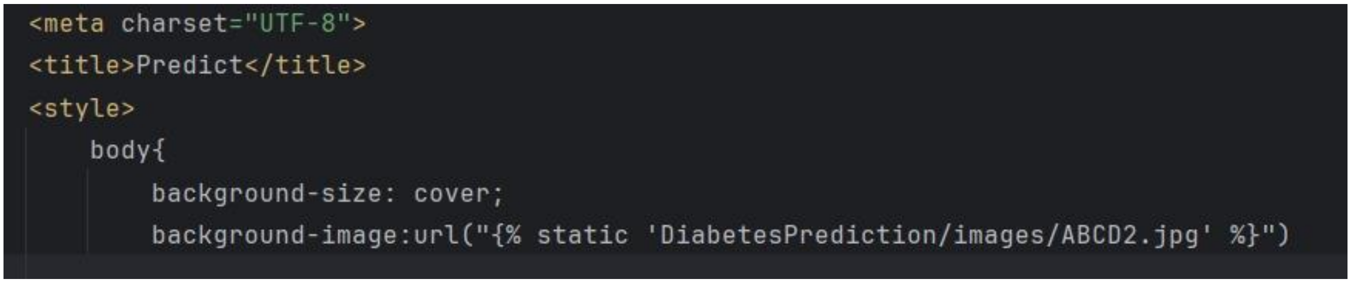


```
    body{
    background-image:url("{% static 'DiabetesPrediction/images/Picture.jpg'%}")

    }
input[type=submit]{
    background-color: #ff4d6b;
```

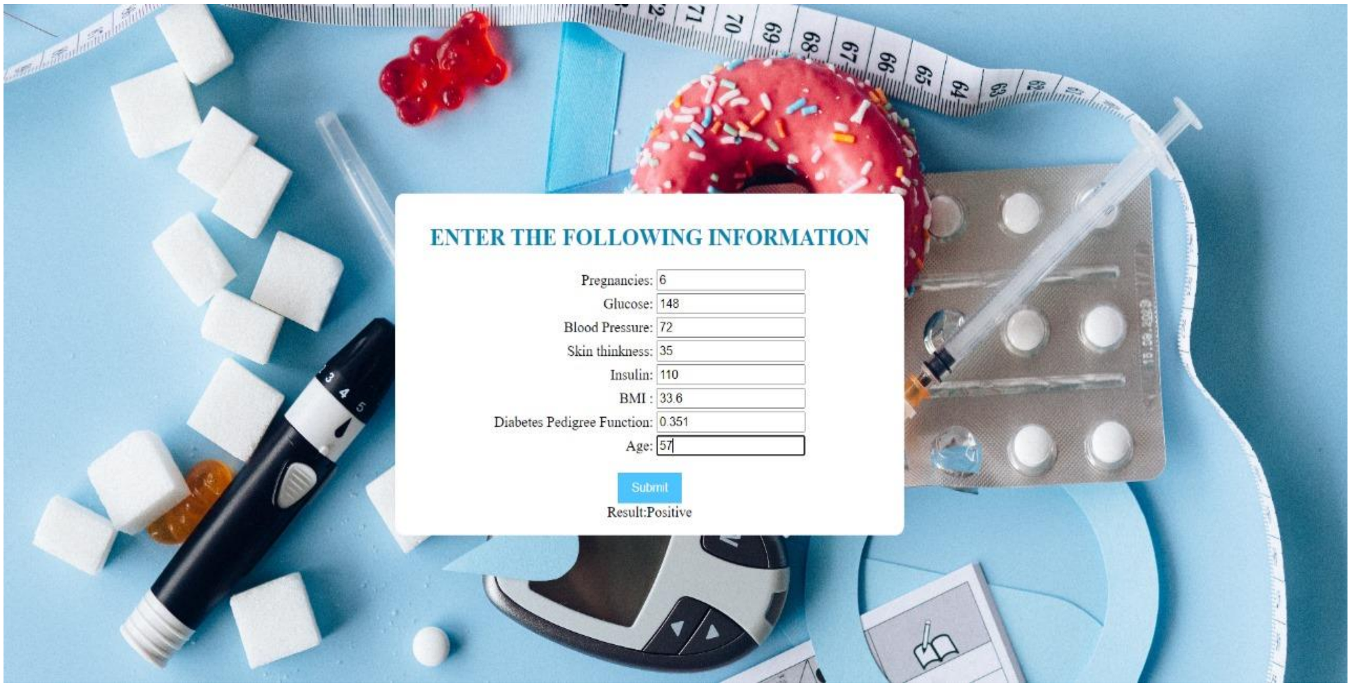


predict.html

- This file consist of the design for the Prediction page
- By right clicking on the image → copy path/reference, you will get the path and can paste it here



```
<meta charset="UTF-8">
<title>Predict</title>
<style>
    body{
        background-size: cover;
        background-image:url("{% static 'DiabetesPrediction/images/ABCD2.jpg' %}")
```



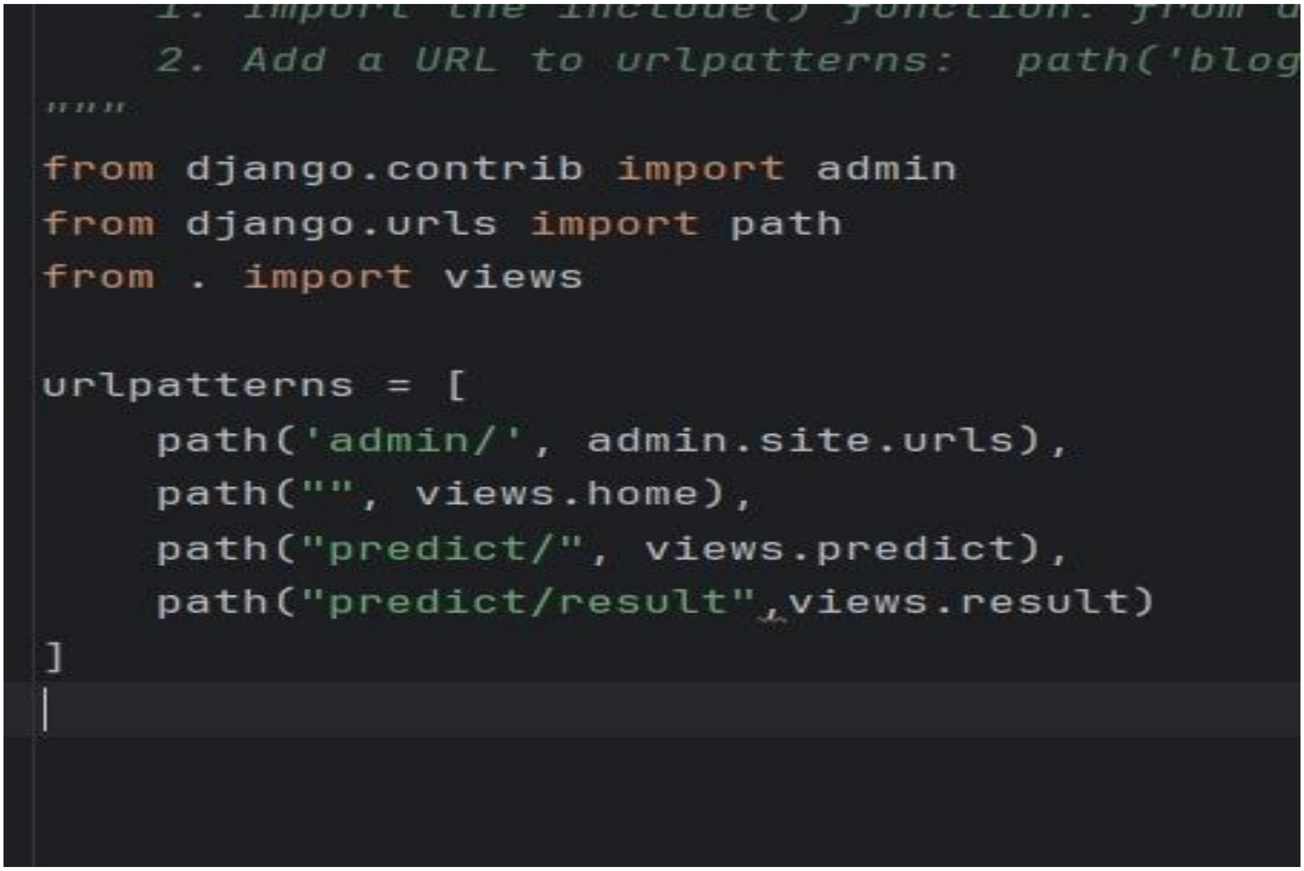
ENTER THE FOLLOWING INFORMATION

Pregnancies: 6
Glucose: 148
Blood Pressure: 72
Skin thinkness: 35
Insulin: 110
BMI : 33.6
Diabetes Pedigree Function: 0.351
Age: 57

Result:Positive

urls.py

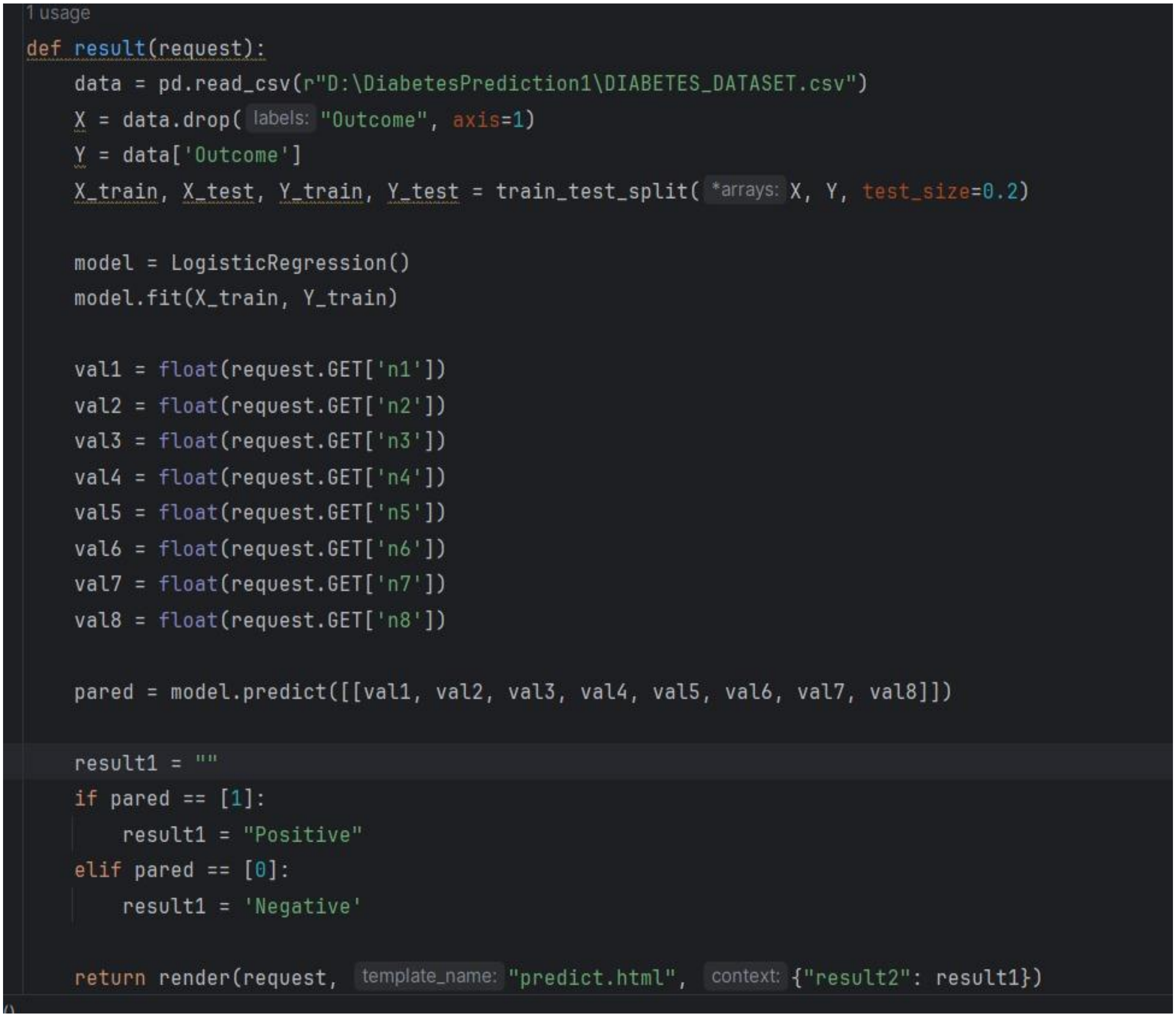
- The urls page must consist of these code.
- It shows the path of the home page, predict page and result



```
    1. Import the include() function: from d
    2. Add a URL to urlpatterns:  path('blog
"""
from django.contrib import admin
from django.urls import path
from . import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path("", views.home),
    path("predict/", views.predict),
    path("predict/result",views.result)
]
```

views.py

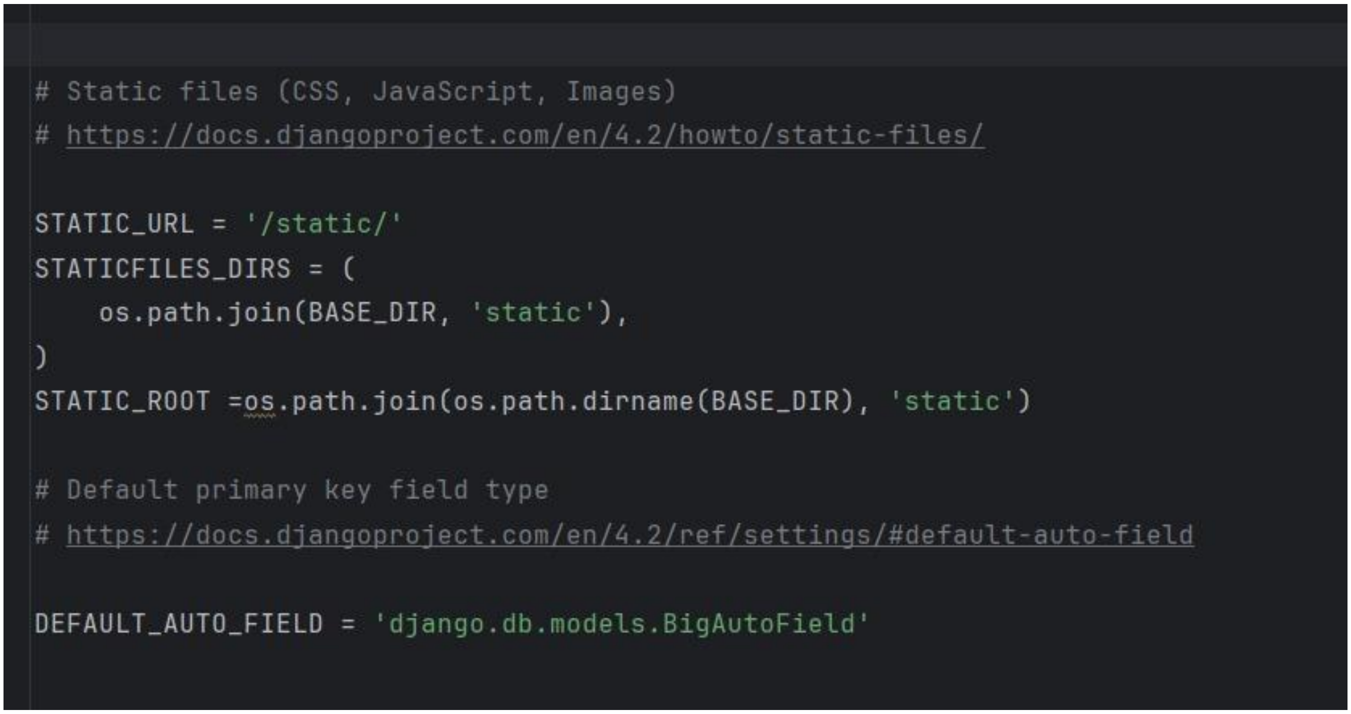
- Copy paste the given code. Change the path of the csv file according to the path in your system where the dataset is stored.This file w ill have the part of getting the input from the user, doing, the prediction and the result will be produced here.



```
1 usage
def result(request):
    data = pd.read_csv(r"D:\DiabetesPrediction1\DIABETES_DATASET.csv")
    X = data.drop( labels: "Outcome", axis=1)
    Y = data['Outcome']
    X_train, X_test, Y_train, Y_test = train_test_split( *arrays: X, Y, test_size=0.2)

    model = LogisticRegression()
    model.fit(X_train, Y_train)

    val1 = float(request.GET['n1'])
    val2 = float(request.GET['n2'])
    val3 = float(request.GET['n3'])
    val4 = float(request.GET['n4'])
    val5 = float(request.GET['n5'])
    val6 = float(request.GET['n6'])
    val7 = float(request.GET['n7'])
    val8 = float(request.GET['n8'])

    pared = model.predict([[val1, val2, val3, val4, val5, val6, val7, val8]])

    result1 = ""
    if pared == [1]:
        result1 = "Positive"
    elif pared == [0]:
        result1 = 'Negative'

    return render(request, template_name: "predict.html", context: {"result2": result1})
```

setting.py

- Edit the settings.py file, at the bottom edit with the following code.



```
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.2/howto/static-files/

STATIC_URL = '/static/'
STATICFILES_DIRS = (
    os.path.join(BASE_DIR, 'static'),
)
STATIC_ROOT =os.path.join(os.path.dirname(BASE_DIR), 'static')

# Default primary key field type
# https://docs.djangoproject.com/en/4.2/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

STEP 6:

- After making all the above changes go to terminal and run the following line of code
- python manage.py runserver.

3. HR ANALYTICS

Tasks to perform

Data Cleaning:

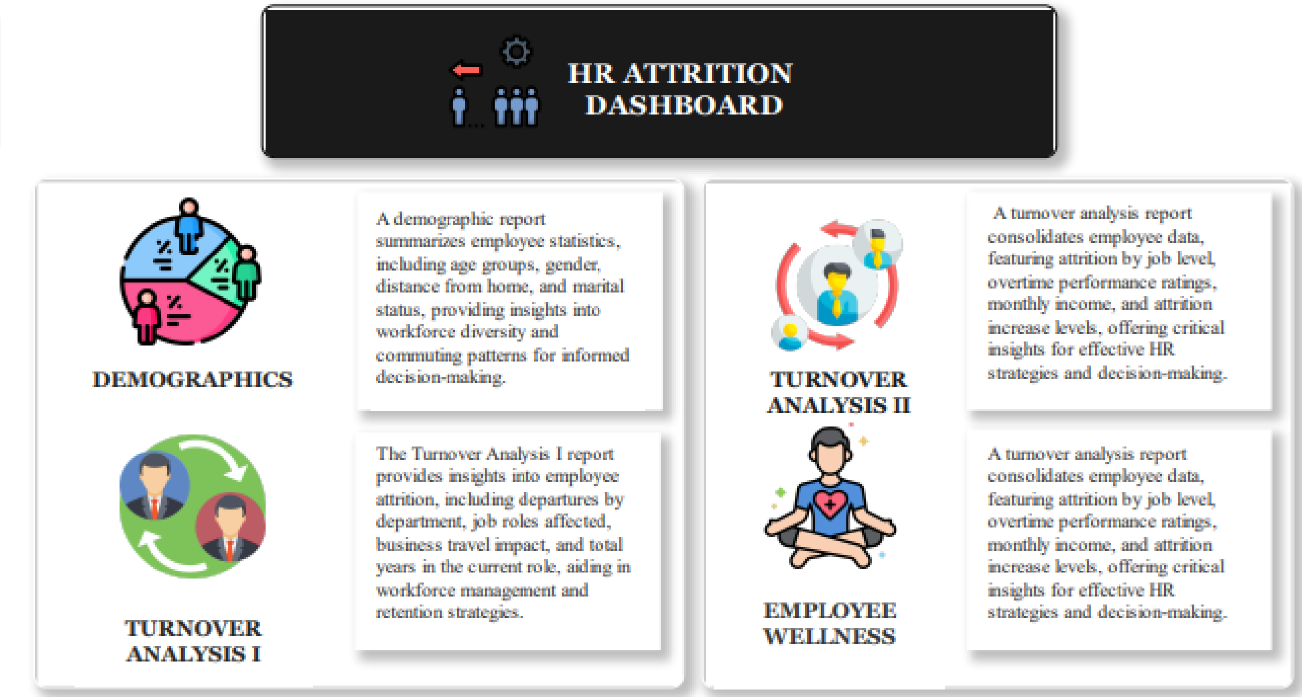
- Deleting redundant columns.
- Renaming the columns.
- Dropping duplicates.
- Cleaning individual columns.
- Remove the NaN values from the dataset
- Check for some more Transformations

Data Visualization:

- Plot a correlation map for all numeric variable Overtime
- Marital Status
- Job Role
- Gender
- Education Field
- Department
- Business Travel
- Relation between Overtime and Age
- Total Working Year
- Education Level
- Number of Companies Worked
- Distance from Home

I downloaded the dataset and uploaded the data using the "Get Data" option in Power BI. Then, I utilized the "Transform" option to check for any null values in the dataset.

PAGE 1:



HR ATTRITION DASHBOARD

DEMOGRAPHICS

A demographic report summarizes employee statistics, including age groups, gender, distance from home, and marital status, providing insights into workforce diversity and commuting patterns for informed decision-making.

TURNOVER ANALYSIS I

The Turnover Analysis I report provides insights into employee attrition, including departures by department, job roles affected, business travel impact, and total years in the current role, aiding in workforce management and retention strategies.

TURNOVER ANALYSIS II

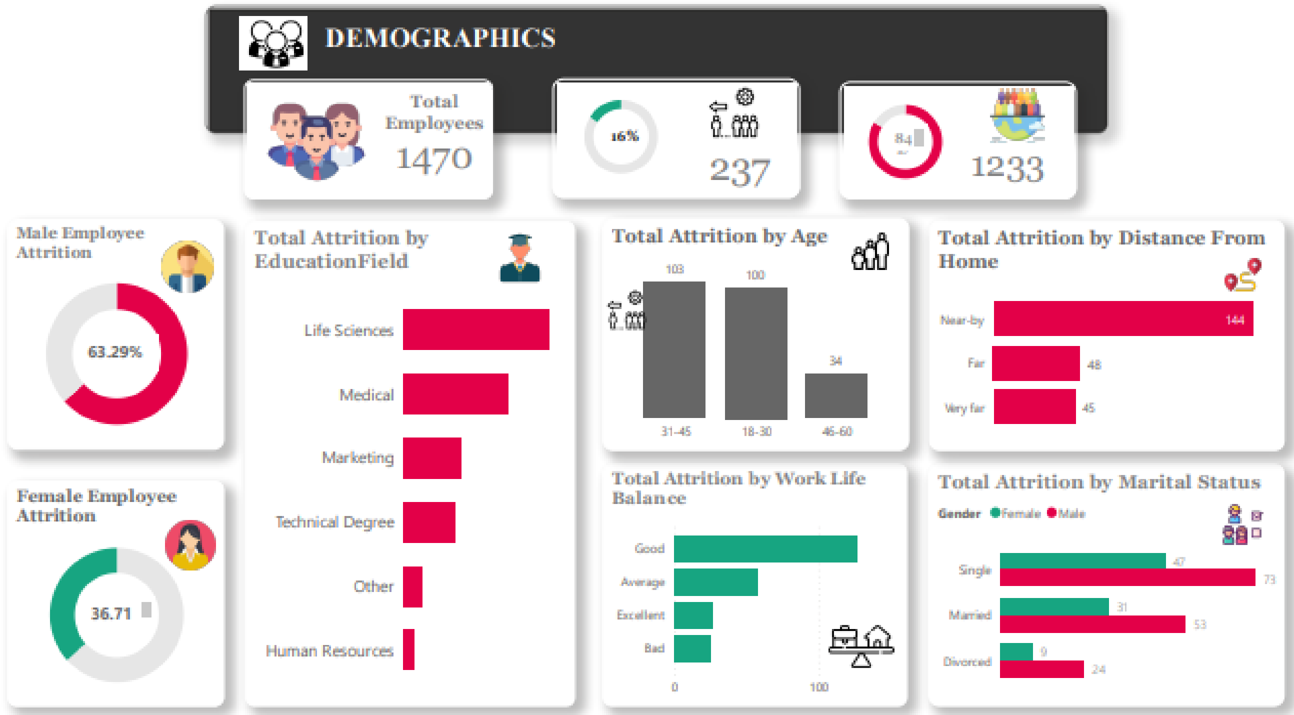
A turnover analysis report consolidates employee data, featuring attrition by job level, overtime performance ratings, monthly income, and attrition increase levels, offering critical insights for effective HR strategies and decision-making.

EMPLOYEE WELLNESS

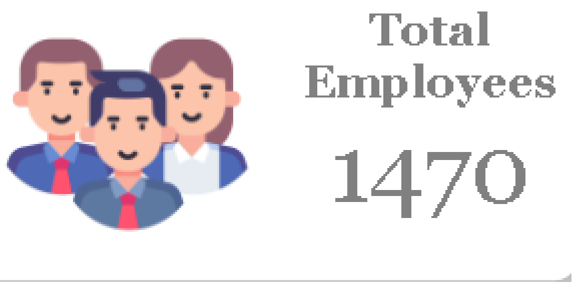
A turnover analysis report consolidates employee data, featuring attrition by job level, overtime performance ratings, monthly income, and attrition increase levels, offering critical insights for effective HR strategies and decision-making.

This first page provides an overview of all the four dashboards I created. The second page includes the demographic insights, the third page focuses on turnover analysis 1, the fourth page on turnover analysis 2, and the final page is dedicated to employee wellness. I will explain the content of each page accordingly.

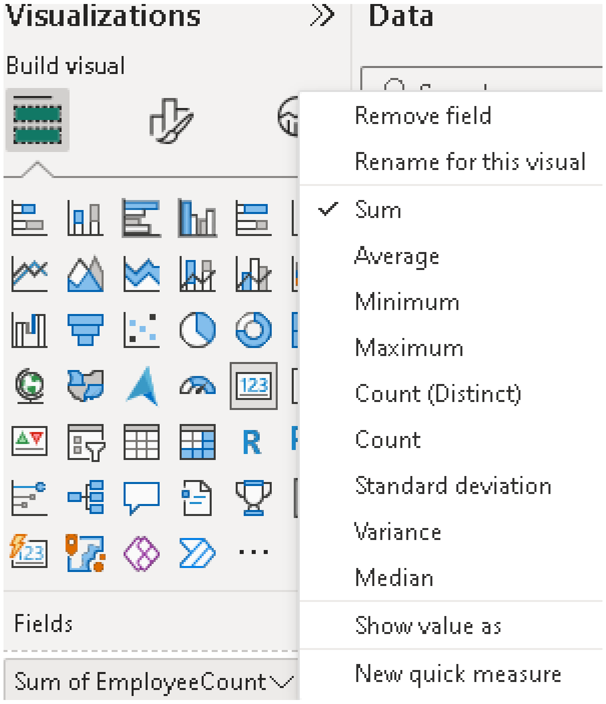
PAGE 2:



STEP 1:



CHOOSE CARD VISUAL

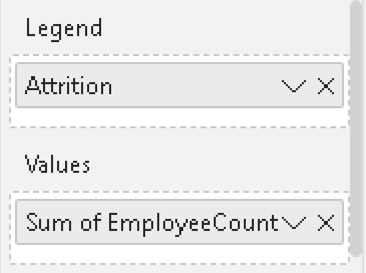


Sum of EmployeeCount

Using the "Format Visual" option, you can customize the design of the card as needed.

STEP 2: Total attrition by education field by clustered bar chart

CHOOSE CLUSTERED BAR CHART VISUAL AND DRAG AND DROP THE BELOW GIVEN COLUMNS

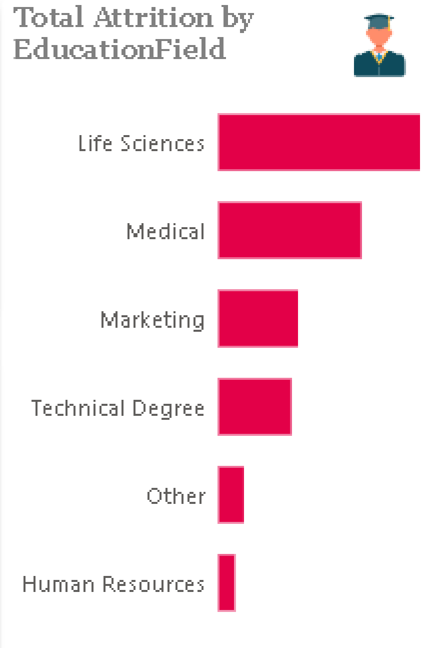


Legend

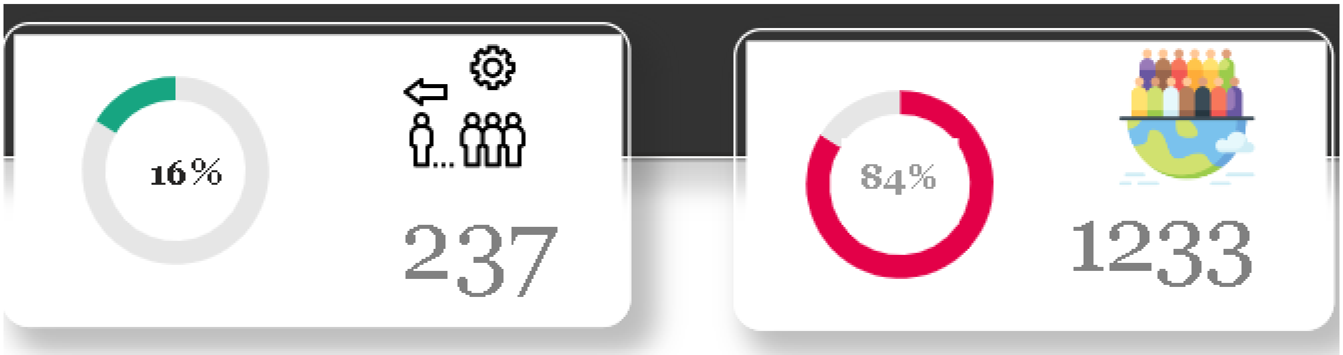
Attrition

Values

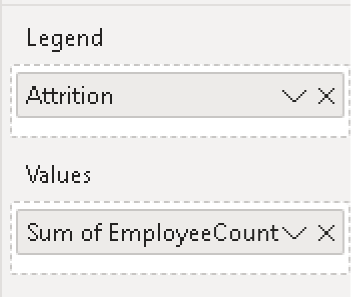
Sum of EmployeeCount



STEP 3:



Choose two donut charts, then drag and drop the provided columns. Apply different colors to distinguish between them.



Legend

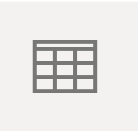
Attrition

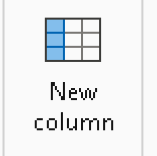
Values

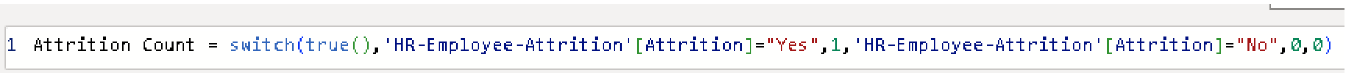
Sum of EmployeeCount

To obtain the counts of 237 and 1233, follow these steps:

- Navigate to the table view, click "New Column," and apply the provided formula.

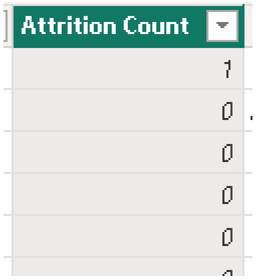




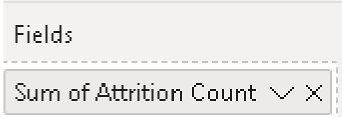


```
Attrition Count = switch(true(),'HR-Employee-Attrition'[Attrition]="Yes",1,'HR-Employee-Attrition'[Attrition]="No",0,0)
```

- This will yield the desired results: if "Attrition" is "Yes," the count is 1; if it's "No," the count is 0.



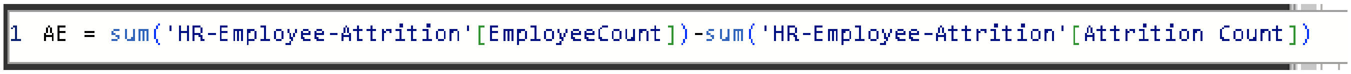
Attrition Count
1
0
0
0
0



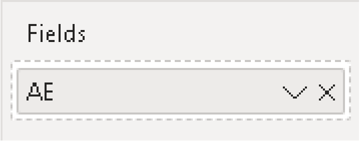
Fields

Sum of Attrition Count

- Summing all the 1s gives the total attrition count that is 237
- And subtracting this from the employee count yields the count of active employees that is 1233



```
AE = sum('HR-Employee-Attrition'[EmployeeCount])-sum('HR-Employee-Attrition'[Attrition Count])
```

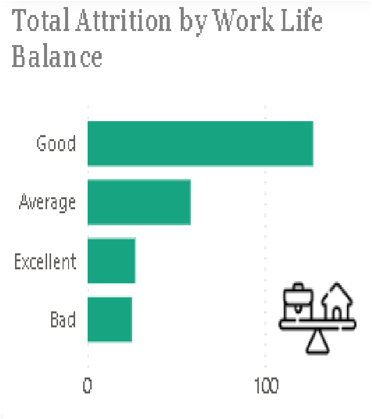


Fields

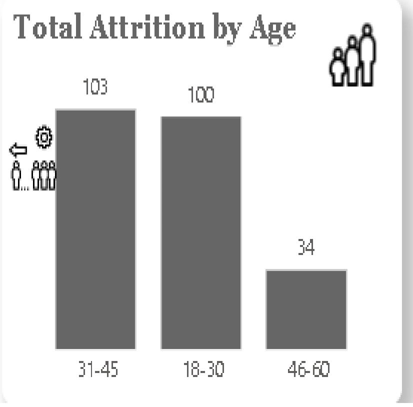
AE

- By dragging and dropping it in the field u will get the active employees

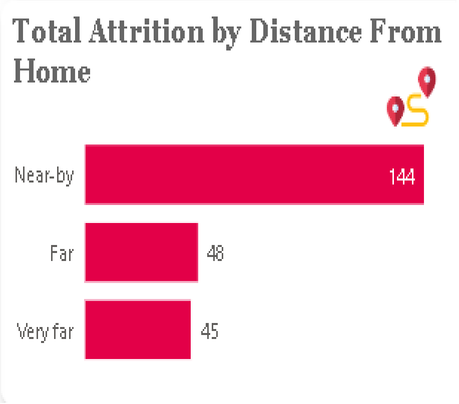
STEP 4: Using data groups for making better visuals. Using groups, you can aggregate certain values together to form meaningful subsets.



Total Attrition by Work Life Balance



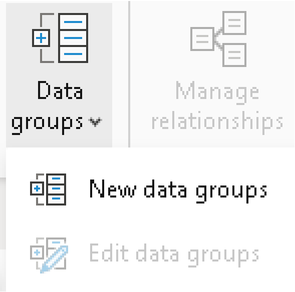
Total Attrition by Age



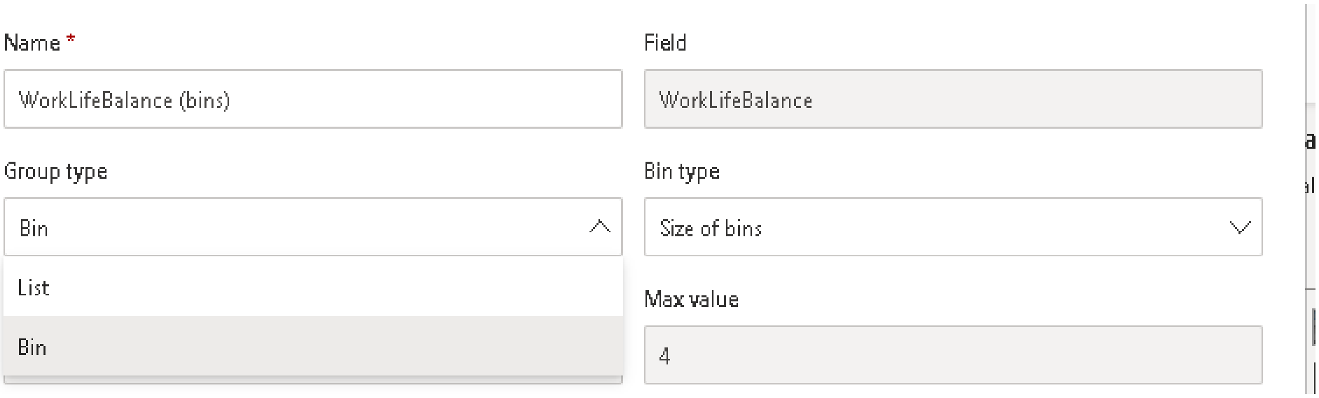
Total Attrition by Distance From Home

We will create these three visuals using the concept of grouping to organize and display the data effectively.

- Choose the column u want to group



- Change the group type from BIN to LIST



Name *
WorkLifeBalance (bins)

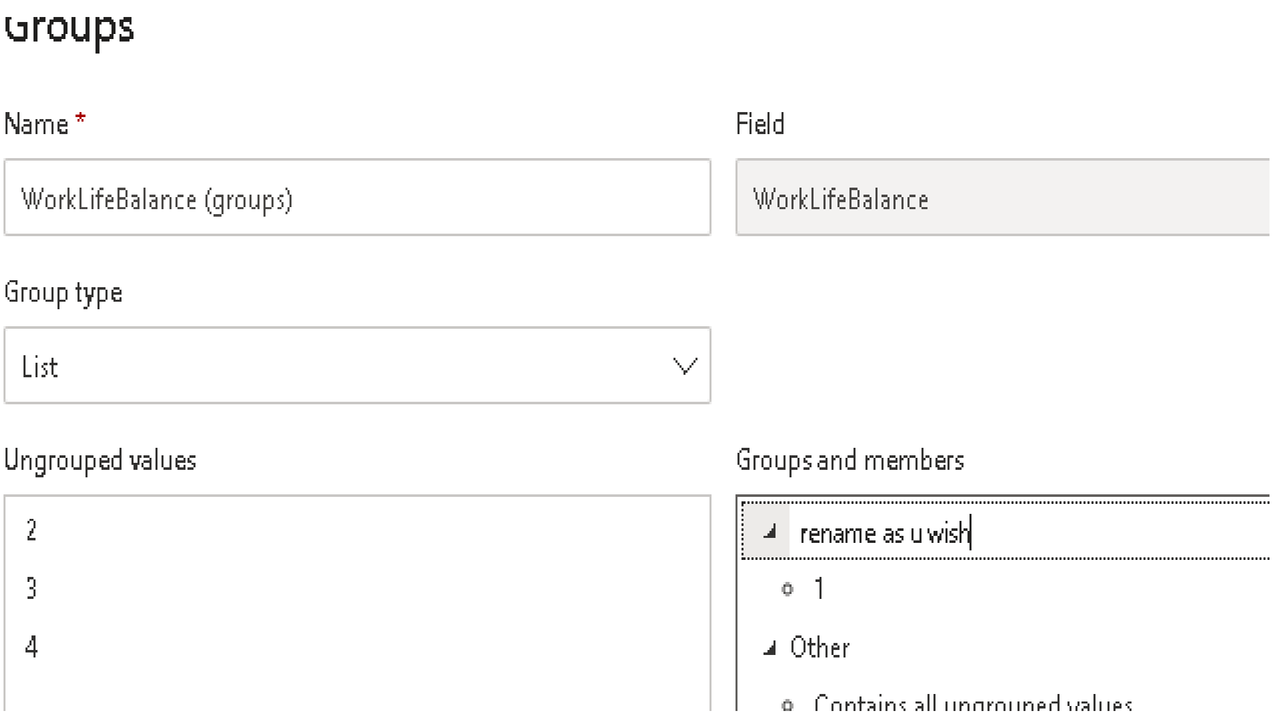
Field
WorkLifeBalance

Group type
Bin

Bin type
Size of bins

Max value
4

- Change From bin to list



Groups

Name *

WorkLifeBalance (groups)

Field

WorkLifeBalance

Group type

List

Ungrouped values

2

3

4

Groups and members

rename as u wish

1

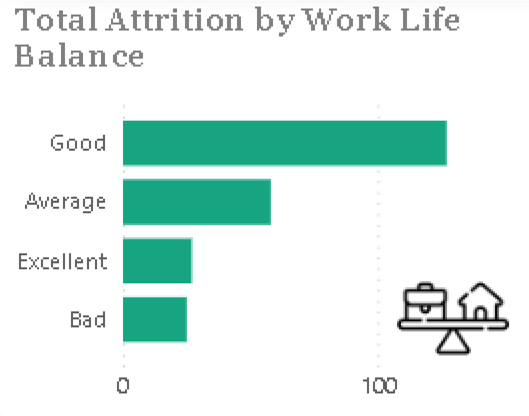
Other

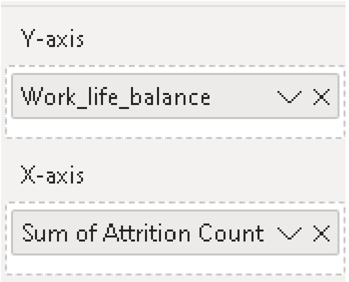
Contains all ungrouped values

- Group 1 as "Bad", 2 as "Average", 3 as "Good", and 4 as "Excellent."
- A new group is created

☑ Work_life_bala...

- Then drag and drop in the fields.





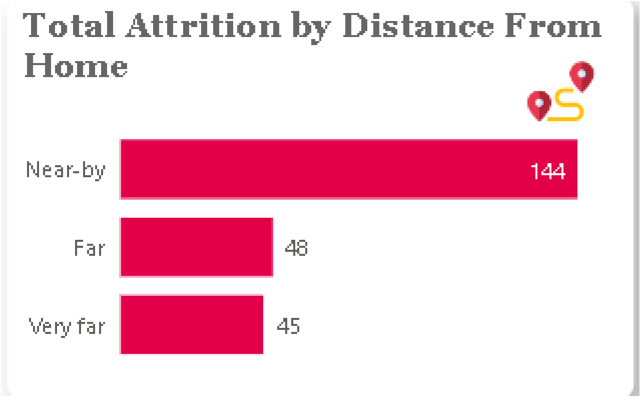
Y-axis

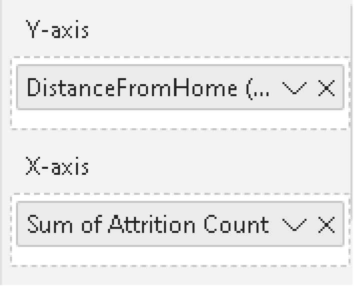
Work_life_balance

X-axis

Sum of Attrition Count

- Follow the same grouping steps for “AGE” and distance from home.
- For distance from home, 1-10 is considered near, 10-20 is considered far and 20 – 30 is considered very far.

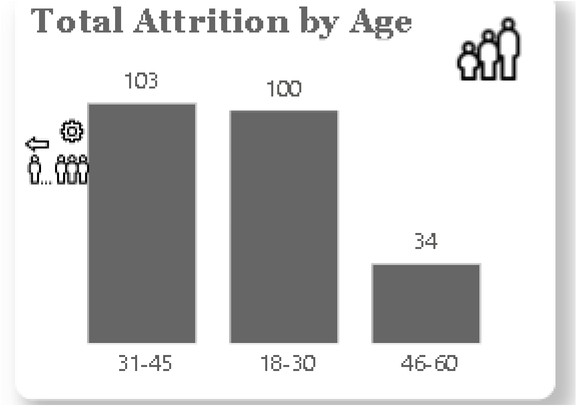


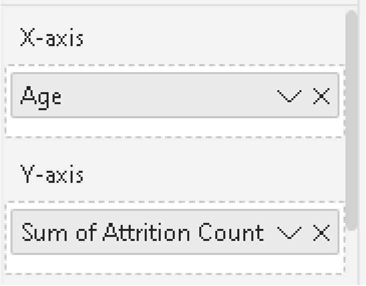

Y-axis

DistanceFromHome (...

X-axis

Sum of Attrition Count



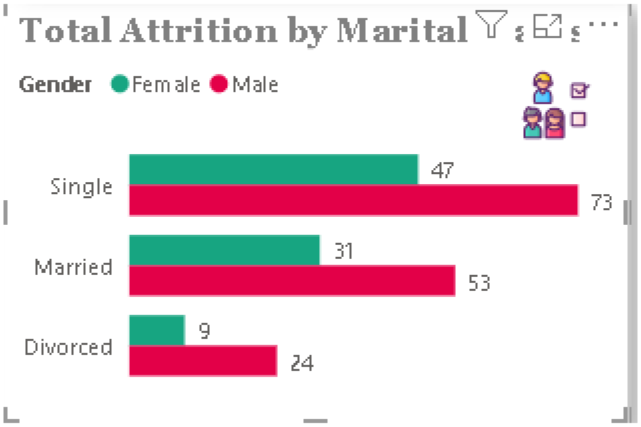

X-axis

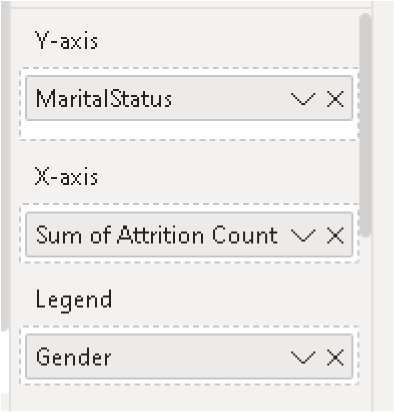
Age

Y-axis

Sum of Attrition Count

STEP 5: Total attrition by marital status using stacked column chart.





Y-axis

MaritalStatus

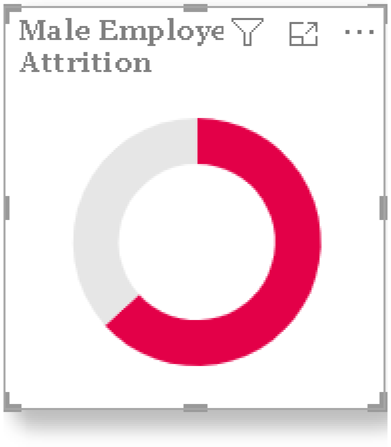
X-axis

Sum of Attrition Count

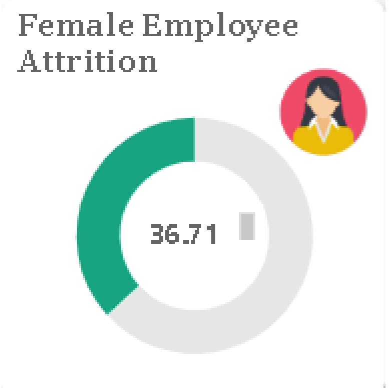
Legend

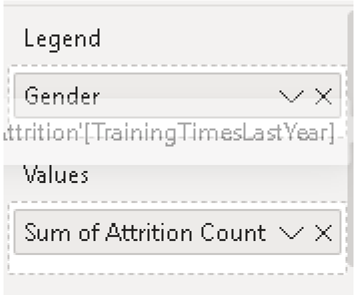
Gender

STEP 6: Gender count using donut chart.



CHOOSE 2 DONUT CHART AND GIVE THE BELOW COLUMNS. THE COLOURS ARE CHANGED TO DIFFERNTIATE





Legend

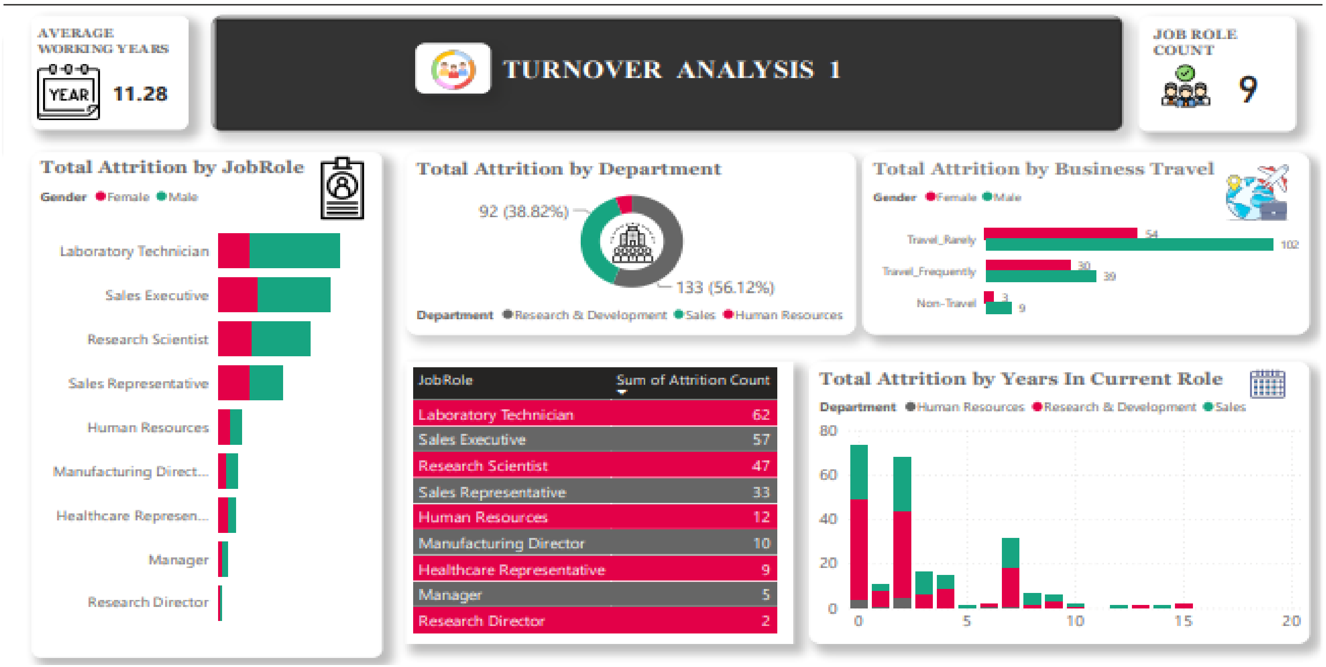
Gender

ttrition'[TrainingTimesLastYear]

Values

Sum of Attrition Count

PAGE 3:



AVERAGE WORKING YEARS 11.28

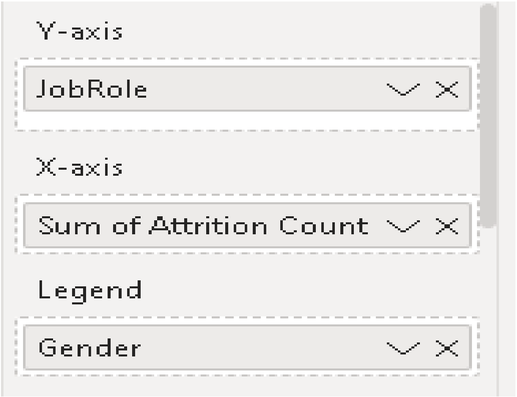
TURNOVER ANALYSIS 1

JOB ROLE COUNT 9

JobRole	Sum of Attrition Count
Laboratory Technician	62
Sales Executive	57
Research Scientist	47
Sales Representative	33
Human Resources	12
Manufacturing Director	10
Healthcare Representative	9
Manager	5
Research Director	2

STEP 1: Total attrition by job role.

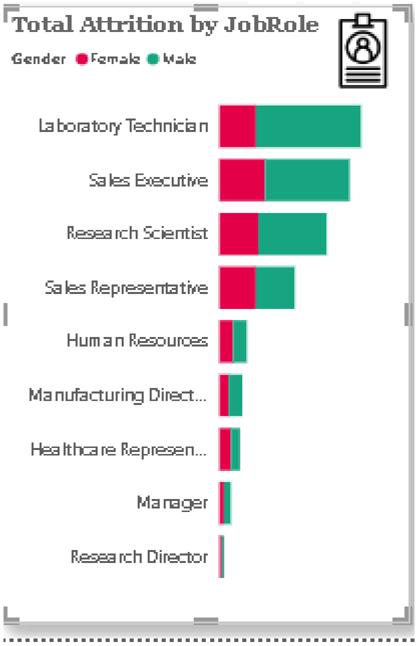
STACKED BAR CHART IS USED HERE



Y-axis
JobRole

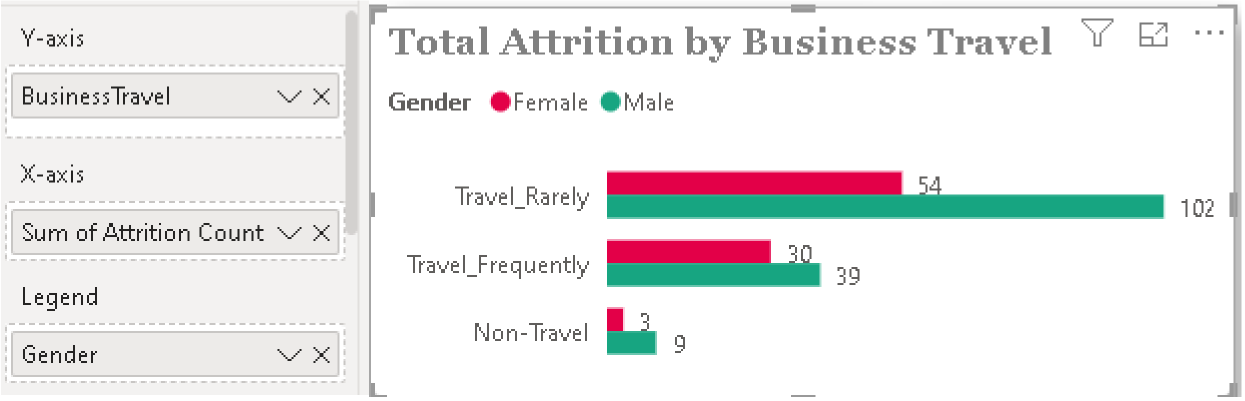
X-axis
Sum of Attrition Count

Legend
Gender



STEP 2:

- **Total attrition by business travel by stacked column chart.**

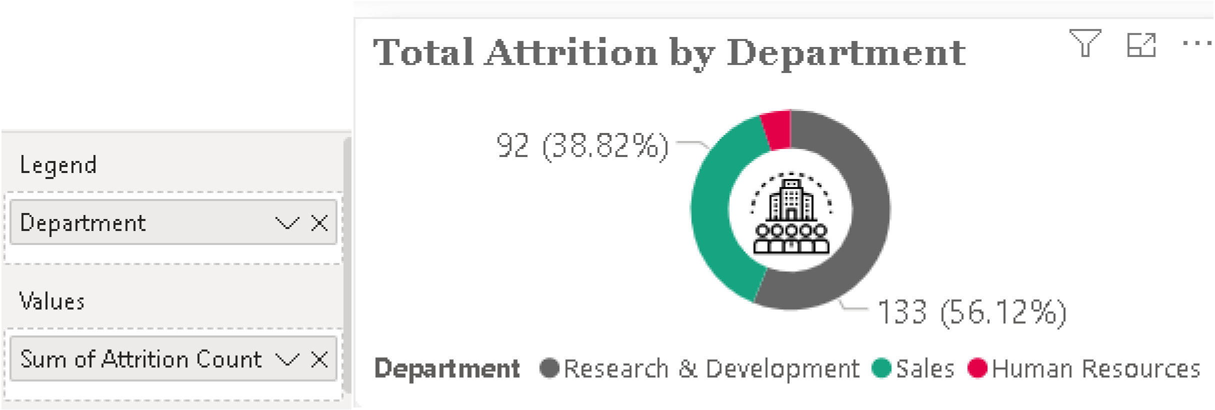


Y-axis
BusinessTravel

X-axis
Sum of Attrition Count

Legend
Gender

- **Total attrition by department using the donut chart**

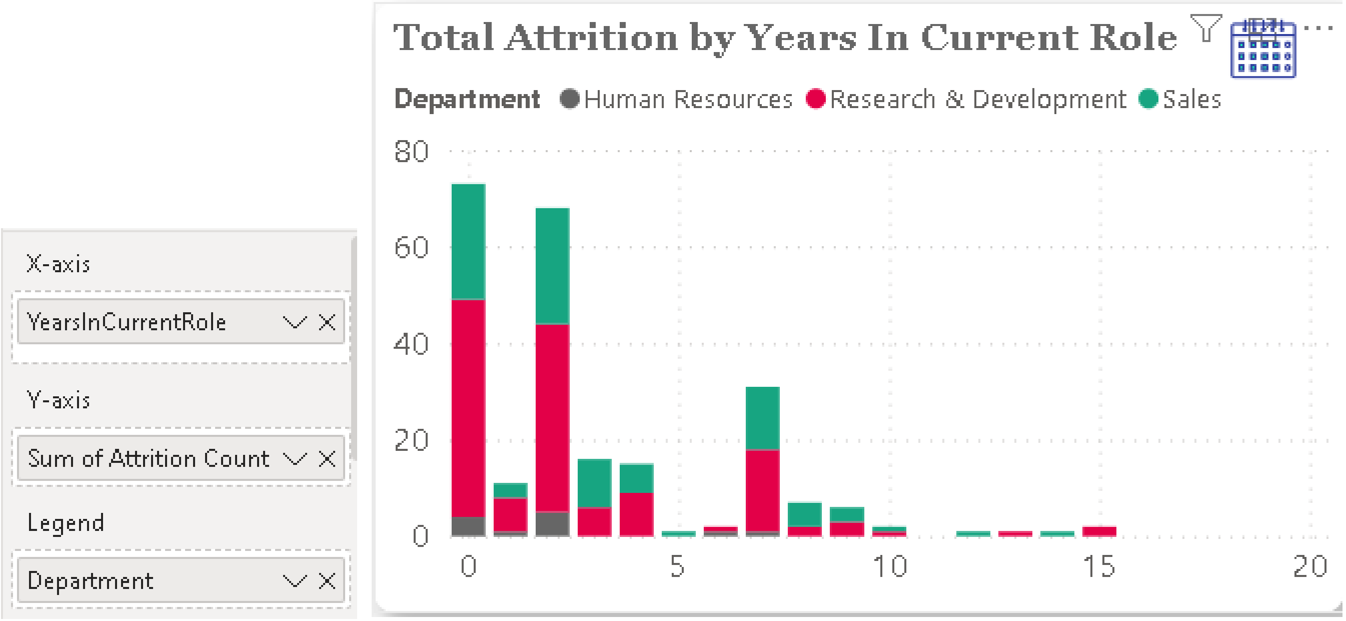


Legend
Department

Values
Sum of Attrition Count

Total Attrition by Department
92 (38.82%)
133 (56.12%)
Department ● Research & Development ● Sales ● Human Resources

- **Total attrition by years in current role using stacked column chart**



X-axis

YearsInCurrentRole

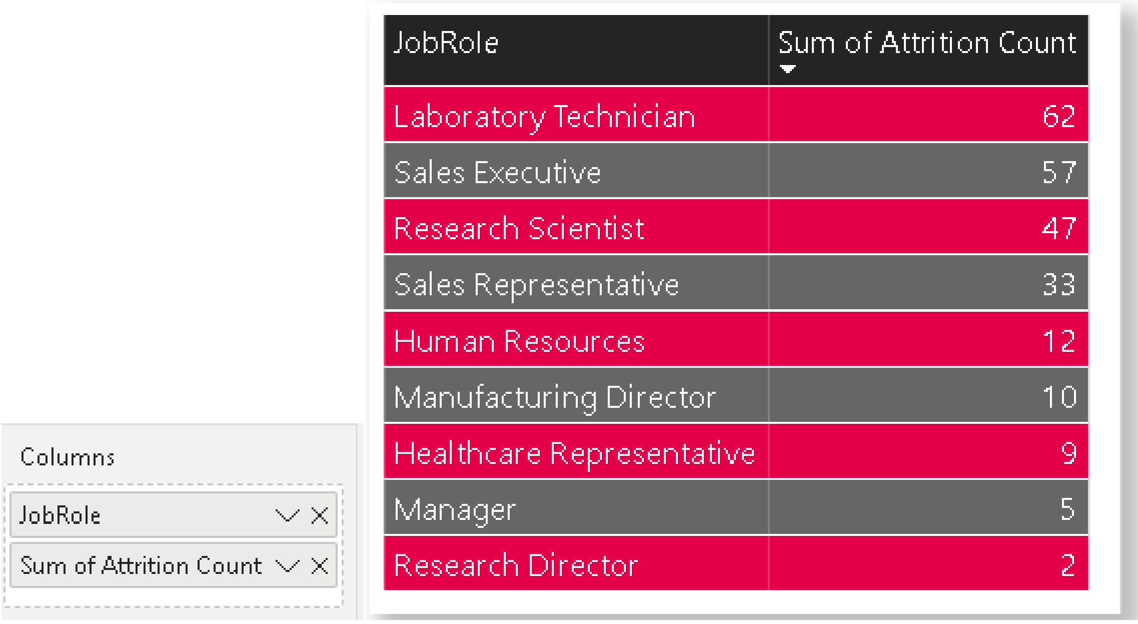
Y-axis

Sum of Attrition Count

Legend

Department

- Total attrition by job role



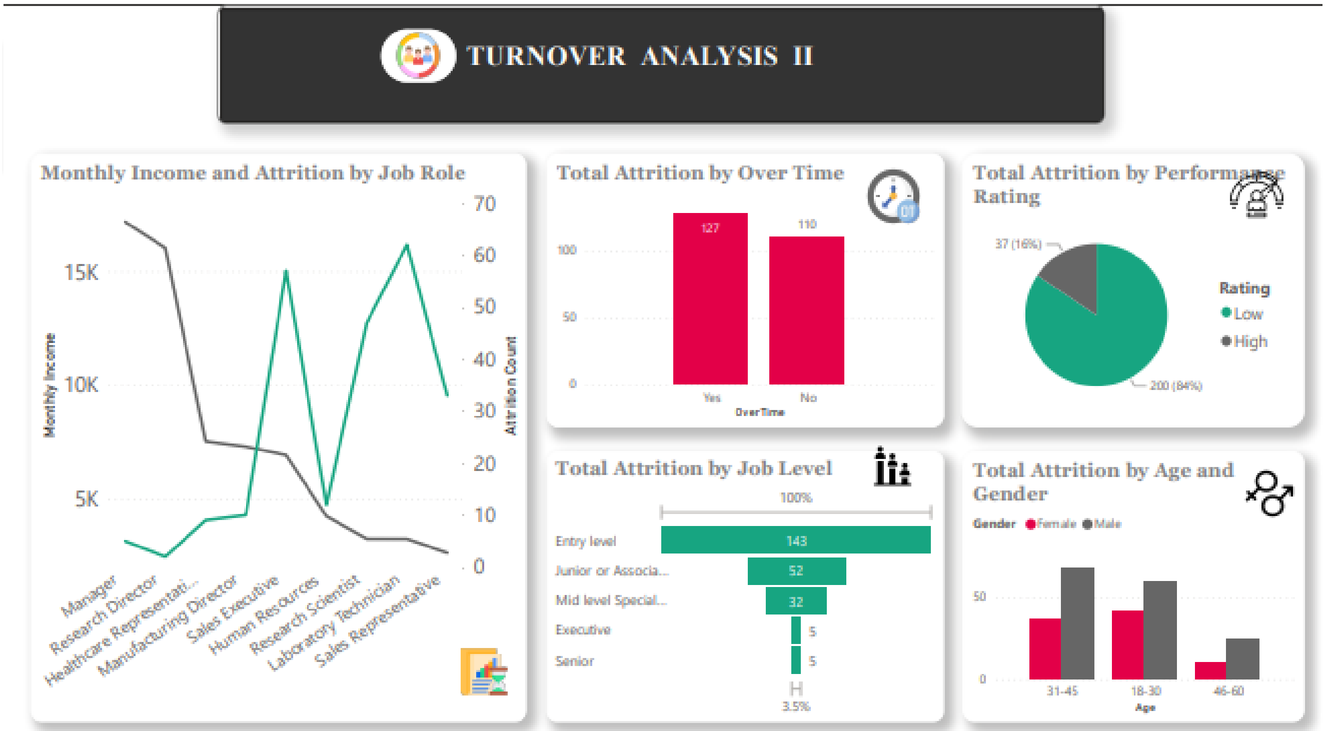
Columns

JobRole

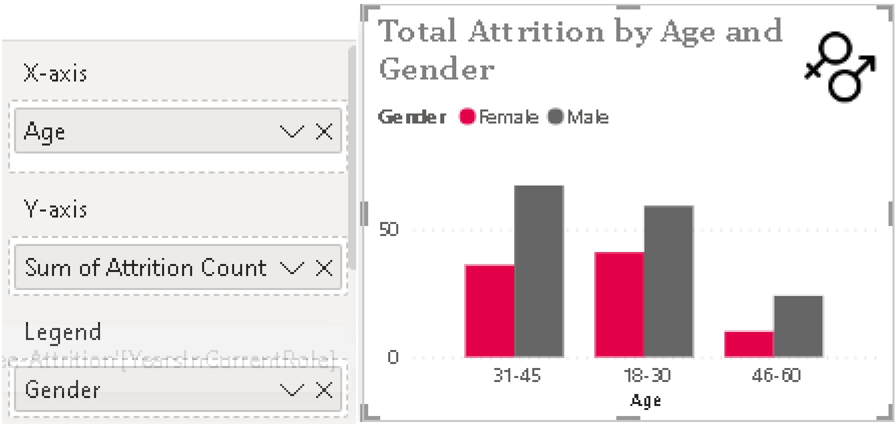
Sum of Attrition Count

JobRole	Sum of Attrition Count
Laboratory Technician	62
Sales Executive	57
Research Scientist	47
Sales Representative	33
Human Resources	12
Manufacturing Director	10
Healthcare Representative	9
Manager	5
Research Director	2

PAGE 4:



- Total attrition by age and gender.



X-axis

Age

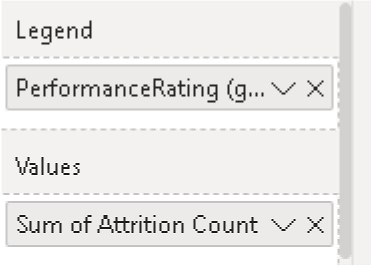
Y-axis

Sum of Attrition Count

Legend

Gender

- Total attrition by performance and rating.

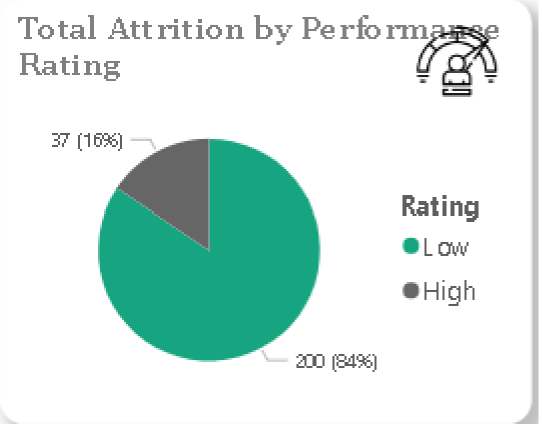


Legend

PerformanceRating (g...

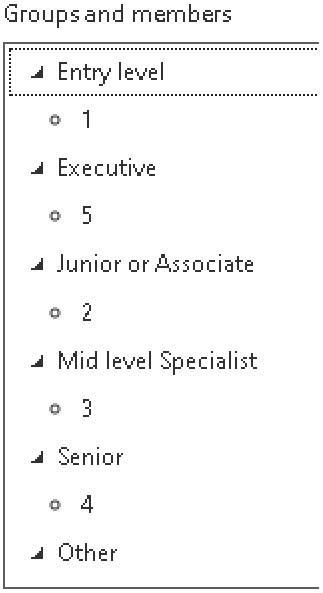
Values

Sum of Attrition Count



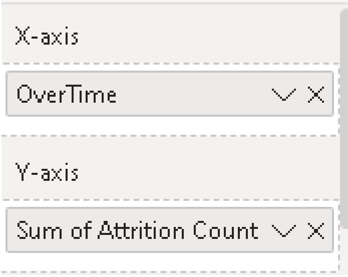
The "Performance Rating" column with values 3 and 4 has been grouped and modified to "Low" and "High" accordingly

- Total attrition by overtime using stacked column chart. Grouping is made as follows



Groups and members

- Entry level
  - 1
- Executive
  - 5
- Junior or Associate
  - 2
- Mid level Specialist
  - 3
- Senior
  - 4
- Other

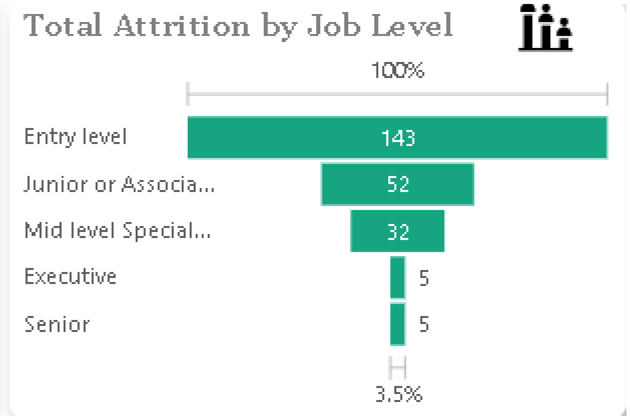


X-axis

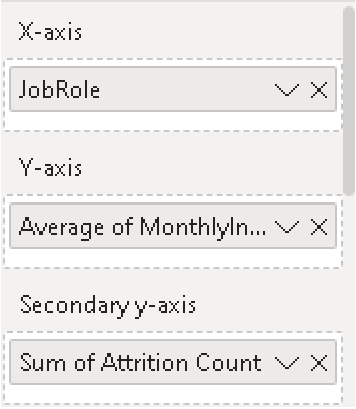
OverTime

Y-axis

Sum of Attrition Count



- Monthly Income and Attrition by job role



X-axis

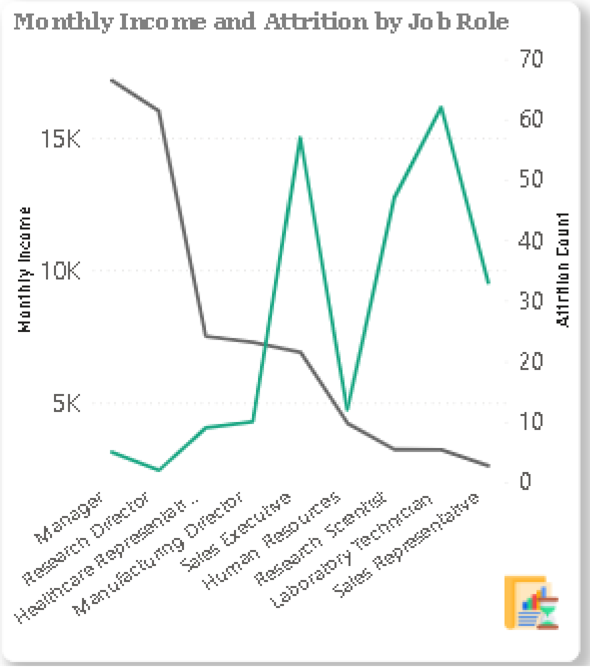
JobRole

Y-axis

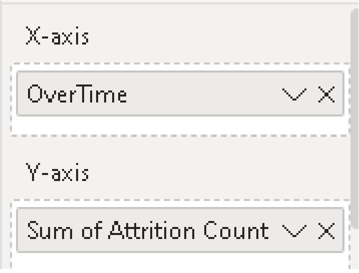
Average of MonthlyIn...

Secondary y-axis

Sum of Attrition Count



- Total attrition by overtime.

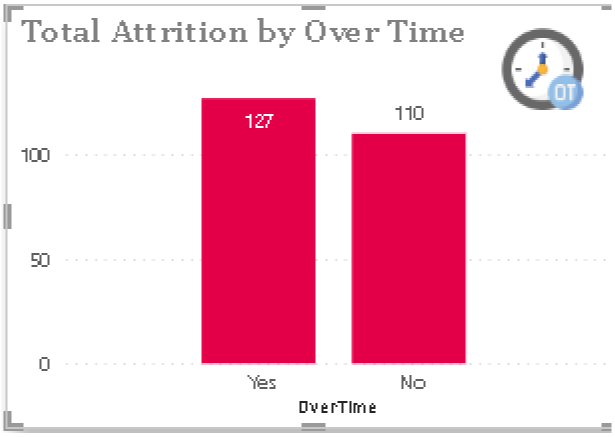


X-axis

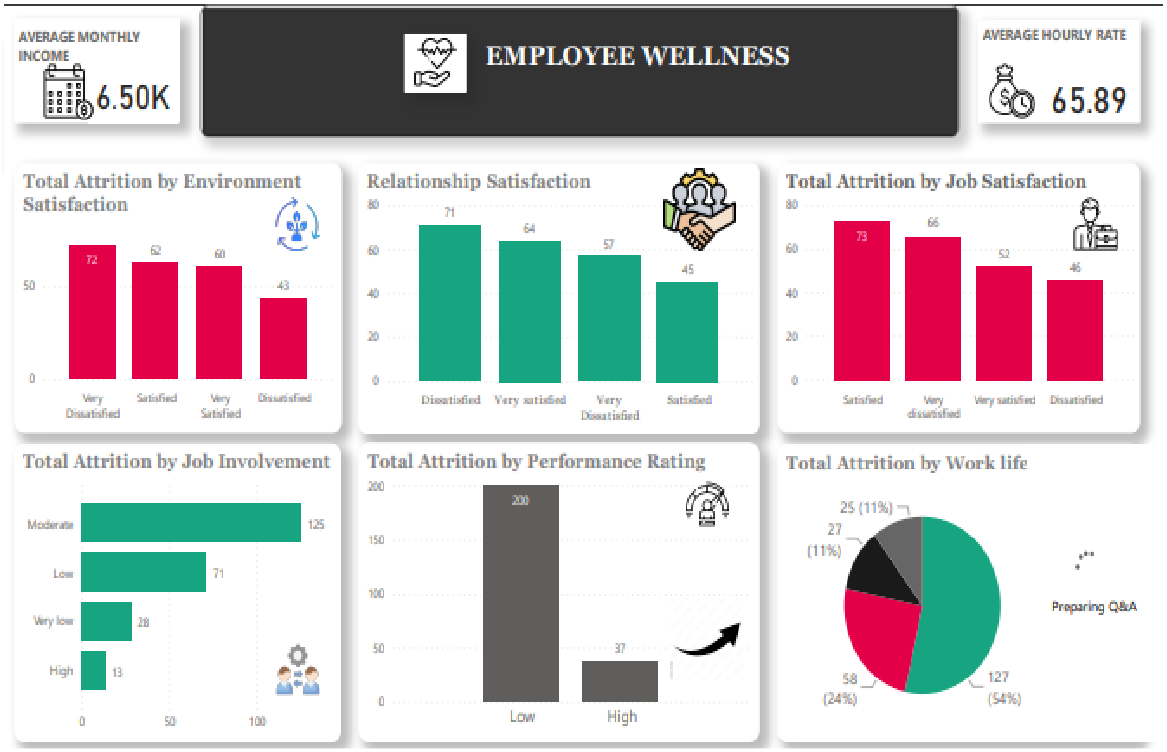
OverTime

Y-axis

Sum of Attrition Count

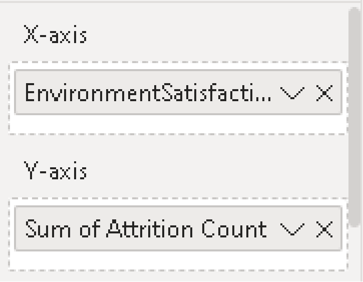


PAGE 5:



STEP 1:

- **Total attrition by environmental satisfaction.**
- **Environmental satisfaction is rated on a scale ranging from 1 to 4.**
- **On the satisfaction scale, 1 represents "very dissatisfied," 2 stands for "dissatisfied," 3 signifies "satisfied," and 4 indicates "very satisfied."**

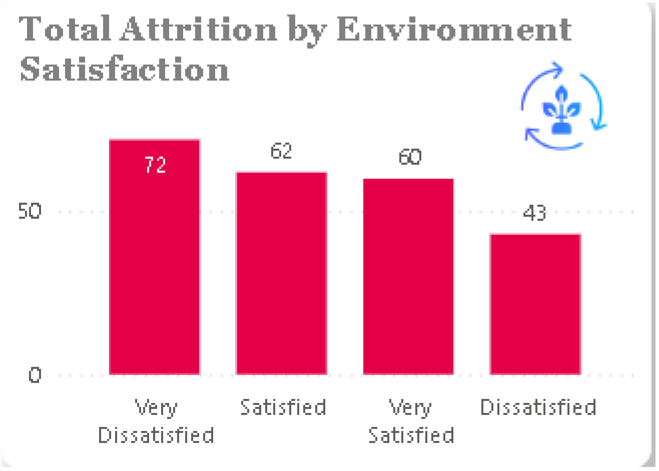


X-axis

EnvironmentSatisfacti... ∨ ✕

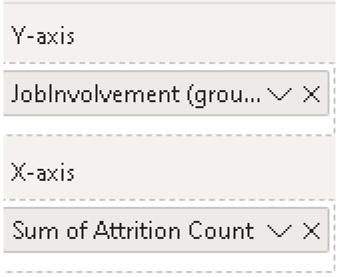
Y-axis

Sum of Attrition Count ∨ ✕



STEP 2:

- Total attrition by job involvement.
- The job involvement is also grouped as very low, low, Moderate, High.

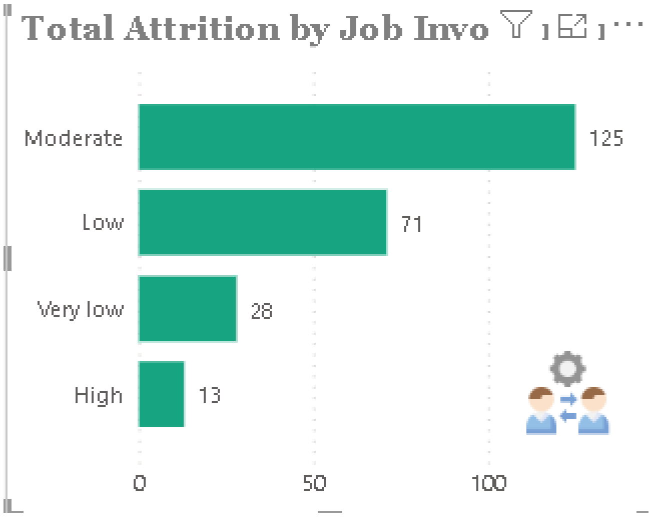


Y-axis

JobInvolvement (grou... ∨ ✕

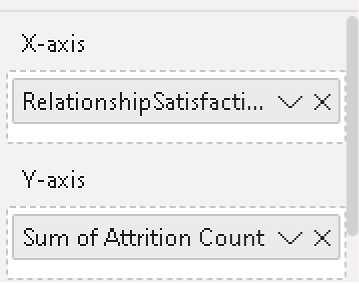
X-axis

Sum of Attrition Count ∨ ✕



STEP 3:

- Relationship satisfaction is grouped as dissatisfied, very dissatisfied, satisfied,very satisfied.

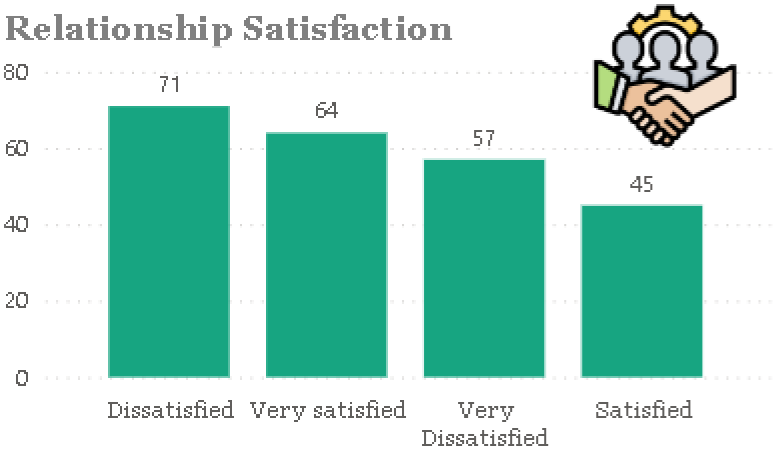


X-axis

RelationshipSatisfacti...

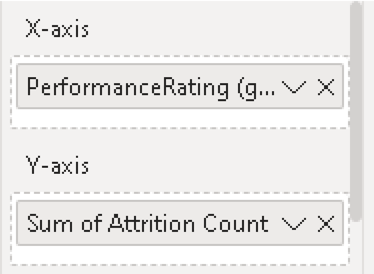
Y-axis

Sum of Attrition Count



STEP 4:

- Performance rating is grouped as low and high.

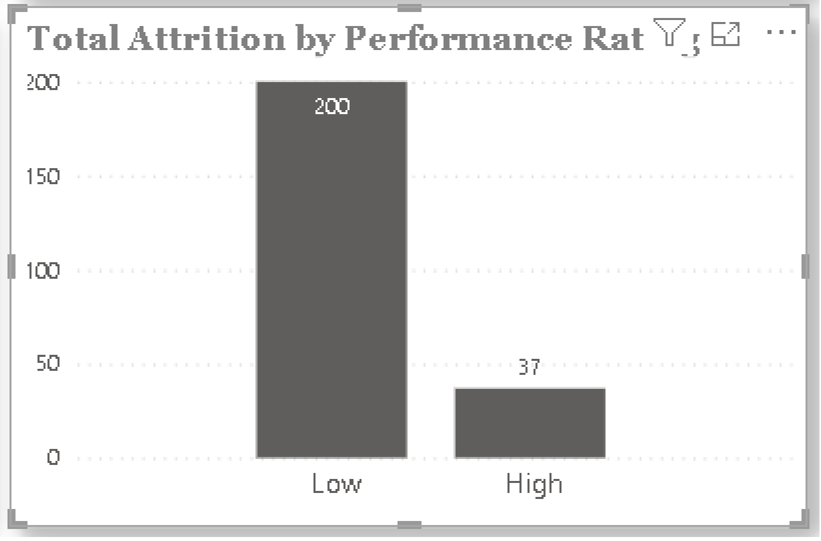


X-axis

PerformanceRating (g...

Y-axis

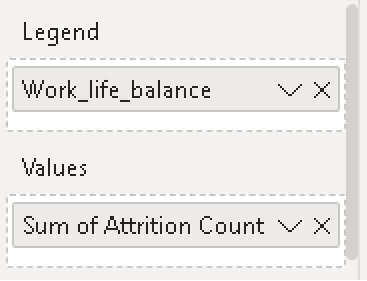
Sum of Attrition Count



STEP 5:

- Total attrition by work life balance.

- On the work life balance, 1 represents "Bad" 2 stands for "Average" 3 signifies "Good" and 4 indicates "Excellent".

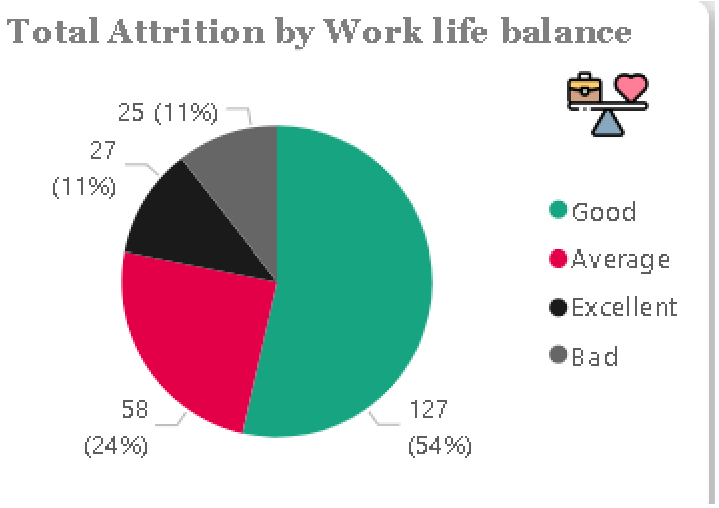


Legend

Work_life_balance

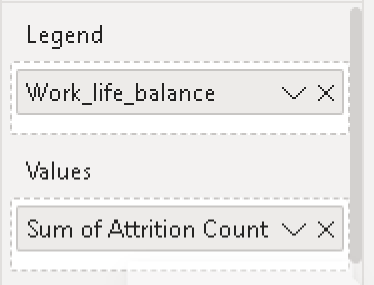
Values

Sum of Attrition Count



STEP 6:

- On the satisfaction scale, 1 represents "very dissatisfied" 2 stands for "dissatisfied" 3 signifies "satisfied" and 4 indicates "very satisfied".

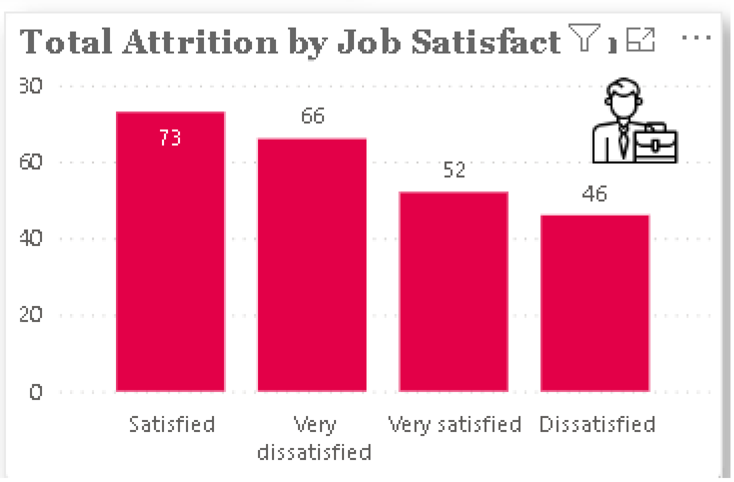


Legend

Work_life_balance

Values

Sum of Attrition Count



In this dashboard, grouping plays a crucial role in organizing and presenting the data effectively. The design of the dashboard is tailored to utilize the power of grouping for enhanced data visualization and analysis.

GUIDE AUTHOR: SAKTHISOWMYA T, TAMIL NADU, INDIA.

AUTHOR'S LINKED IN: https://www.linkedin.com/in/sakthisowmya-t-97925922a/.