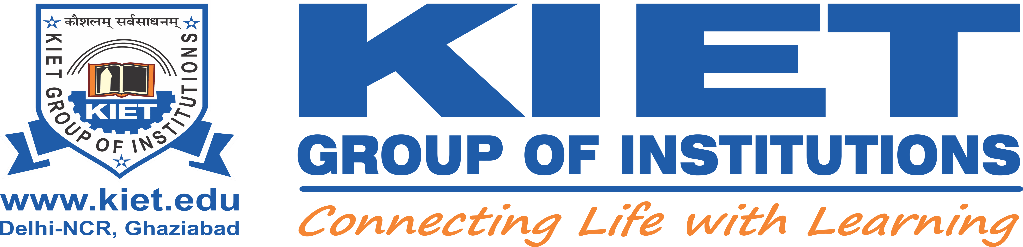
****

**Assessment Report**

on

**“Predict Disease Outcome Based on Genetic and Clinical Data”**

submitted as partial fulfilment for the award of

**BACHELOR OF TECHNOLOGY**

**DEGREE**

SESSION 2024-25

in

**Name of discipline**

By

Prajesh Singh Meena (202401100400136)

**Under the supervision of**

“Abhishek Shukla”

**KIET Group of Institutions, Ghaziabad**

Affiliated to

**Dr. A.P.J. Abdul Kalam Technical University, Lucknow**

(Formerly UPTU)

**May, 2025**

1. **Problem Statement**

In the era of data science and machine learning, data visualization is a crucial step in understanding patterns and making informed decisions. This project involves uploading a real-world dataset, analysing it using data visualization techniques, and identifying relationships between different numerical variables through a correlation heatmap. The objective is to gain insights into the data structure and highlight which features are strongly correlated.

We will use Python as our primary programming language due to its simplicity and powerful data analysis libraries such as pandas, matplotlib, and seaborn. The goal is to process the dataset from scratch, clean it if necessary, generate histograms to visualize distributions, and finally, produce a heatmap that reflects the correlations between features.

**2. Introduction**

With the explosion of data in every field, making sense of that data is more important than ever. Raw numbers are difficult to interpret, especially in large datasets. This is where data visualization steps in. Visualization tools help to quickly grasp patterns, trends, and outliers that might otherwise go unnoticed. In machine learning and data analysis, visualizations not only enhance our understanding but also guide decision-making about which features to use or remove.

In this project, we are working with a dataset (most likely the Breast Cancer Wisconsin dataset, based on the structure), which contains measurements from cell nuclei present in breast cancer biopsies. Each feature corresponds to a property like radius, texture, perimeter, and so on. Our task is to load this dataset, explore it visually, and understand which features are most closely related to each other.

We will create histograms to observe the distribution of each feature and use a heatmap to display the correlation between every pair of numerical features. This is especially useful in identifying multicollinearity, selecting features for modeling, or simply understanding the dataset better.

**3. Methodology**

conduct this analysis, the following approach was followed:

1. **Dataset Upload**  
   Using Google Colab's files.upload() feature, we uploaded the CSV file containing our dataset. This is a convenient method for handling files when working in cloud-based notebooks.
2. **Data Loading and Exploration**  
   Once the file was uploaded, we used pandas to load the dataset into a DataFrame. We then printed the first few rows using head() and examined its structure using info(). This allowed us to check for any missing values, non-numeric columns, or unnecessary fields like an ID column.
3. **Data Cleaning (if necessary)**  
   In some cases, datasets contain columns that aren't useful for analysis, such as unnamed columns or IDs. These were excluded from the correlation analysis by filtering out columns whose names contained 'id' or 'Unnamed'.
4. **Histogram Plotting**  
   A histogram was generated for each numerical feature using DataFrame.hist() with matplotlib. These visualizations helped us understand the spread, skewness, and possible outliers in the data.
5. **Correlation Matrix and Heatmap**  
   To analyze the relationships between features, we computed the correlation matrix using DataFrame.corr() on numeric columns. We visualized this using a seaborn.heatmap() with annotations, color coding, and proper label formatting to make it visually informative and accessible.
6. **Final Adjustments**  
   The heatmap was formatted with appropriate sizing (figsize), font adjustments (annot\_kws), and layout fixes (tight\_layout()) to ensure readability even for datasets with many features.

**4. Code**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from google.colab import files

def upload\_dataset():

uploaded = files.upload()

if uploaded:

file\_name = next(iter(uploaded))

return file\_name

return None

def load\_and\_display\_data(file\_path):

data = pd.read\_csv(file\_path)

print("Dataset preview:")

print(data.head())

print("\nDataset info:")

data.info()

return data

def visualize\_data(data):

data.hist(bins=20, figsize=(12, 10))

plt.tight\_layout()

plt.show()

numeric\_data = data.select\_dtypes(include=['number'])

numeric\_data = numeric\_data.loc[:, ~numeric\_data.columns.str.contains('^Unnamed|id', case=False)]

correlation\_matrix = numeric\_data.corr()

plt.figure(figsize=(16, 14))

sns.heatmap(

correlation\_matrix,

annot=True,

cmap="coolwarm",

fmt=".2f",

square=True,

annot\_kws={"size": 8}

)

plt.xticks(rotation=45, ha='right', fontsize=10)

plt.yticks(rotation=0, fontsize=10)

plt.title("Correlation Heatmap", fontsize=16, pad=20)

plt.tight\_layout()

plt.show()

def main():

file\_path = upload\_dataset()

if not file\_path:

return

data = load\_and\_display\_data(file\_path)

visualize\_data(data)

main()

plt.figure(figsize=(16, 14))

sns.heatmap(

correlation\_matrix,

annot=True,

cmap="coolwarm",

fmt=".2f",

square=True,

annot\_kws={"size": 8}

)

plt.xticks(rotation=45, ha='right', fontsize=10)

plt.yticks(rotation=0, fontsize=10)

plt.title("Correlation Heatmap", fontsize=16, pad=20)

plt.tight\_layout()

plt.show()

def main():

file\_path = upload\_dataset()

if not file\_path:

return

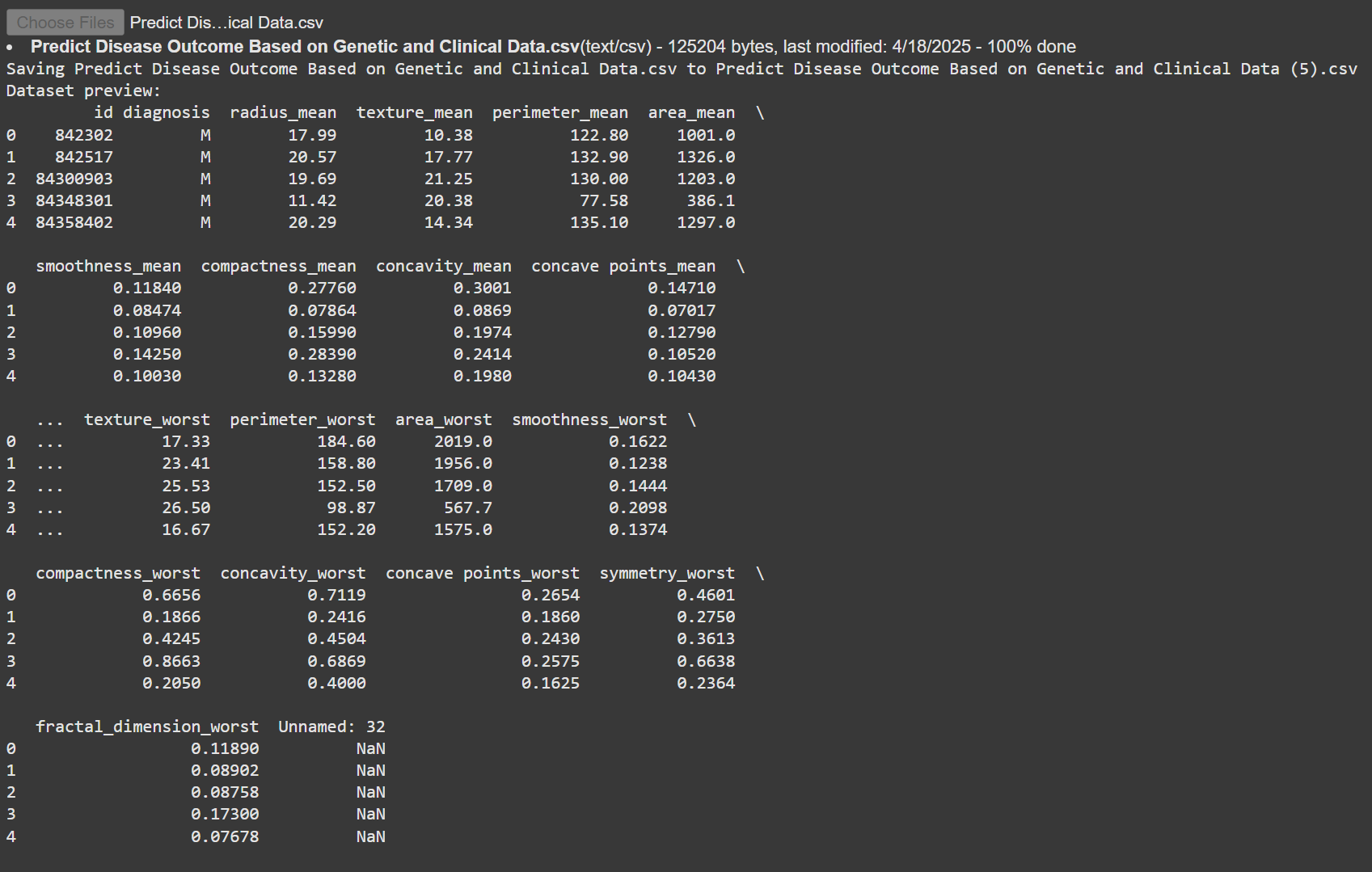
data = load\_and\_display\_data(file\_path)

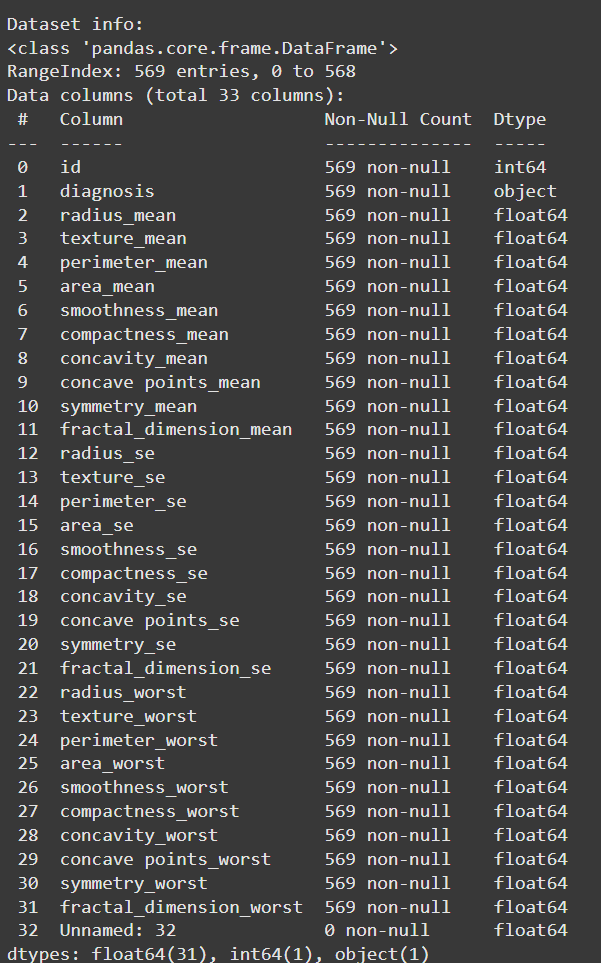
visualize\_data(data)

main()

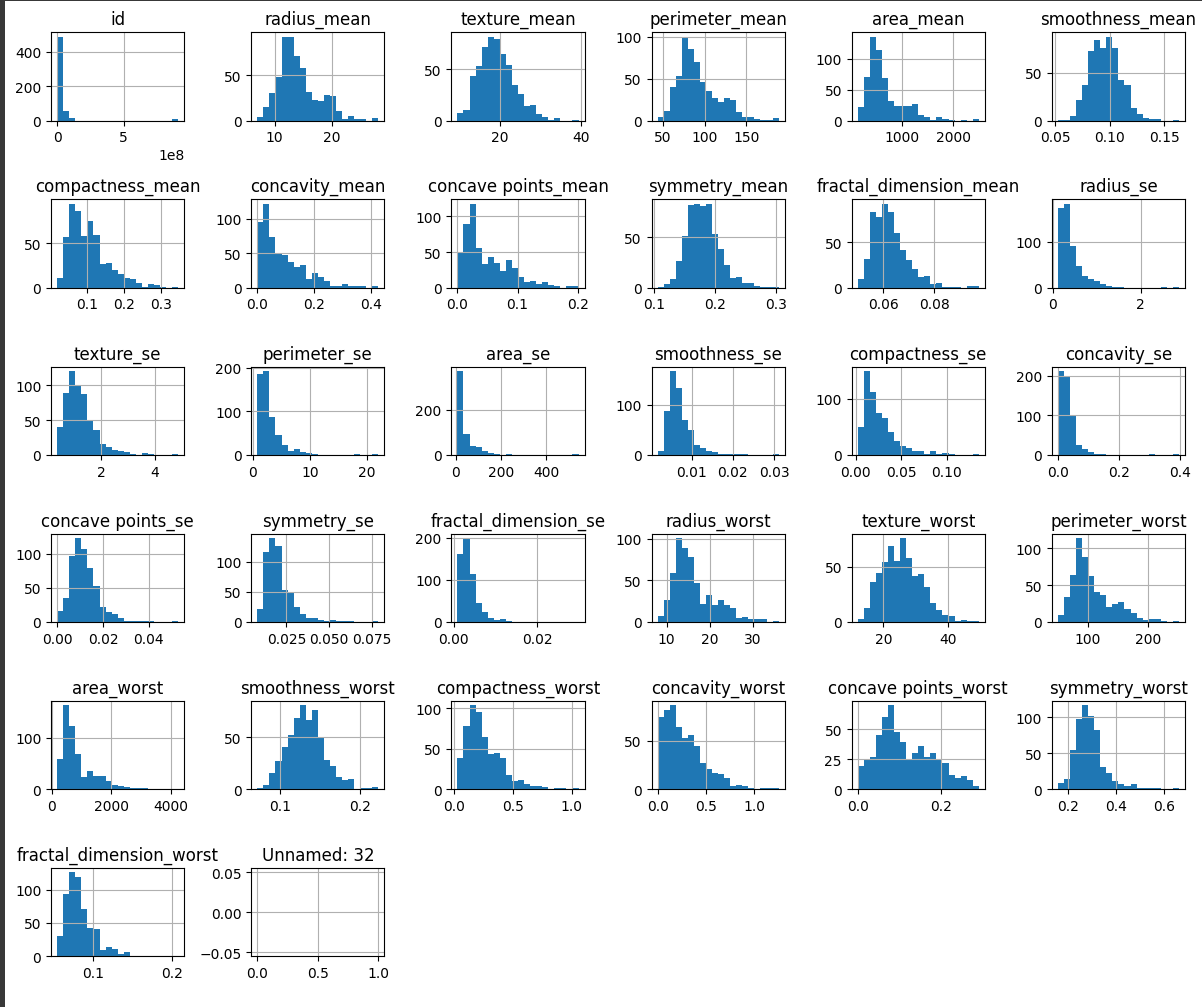
**5. Output / Result**

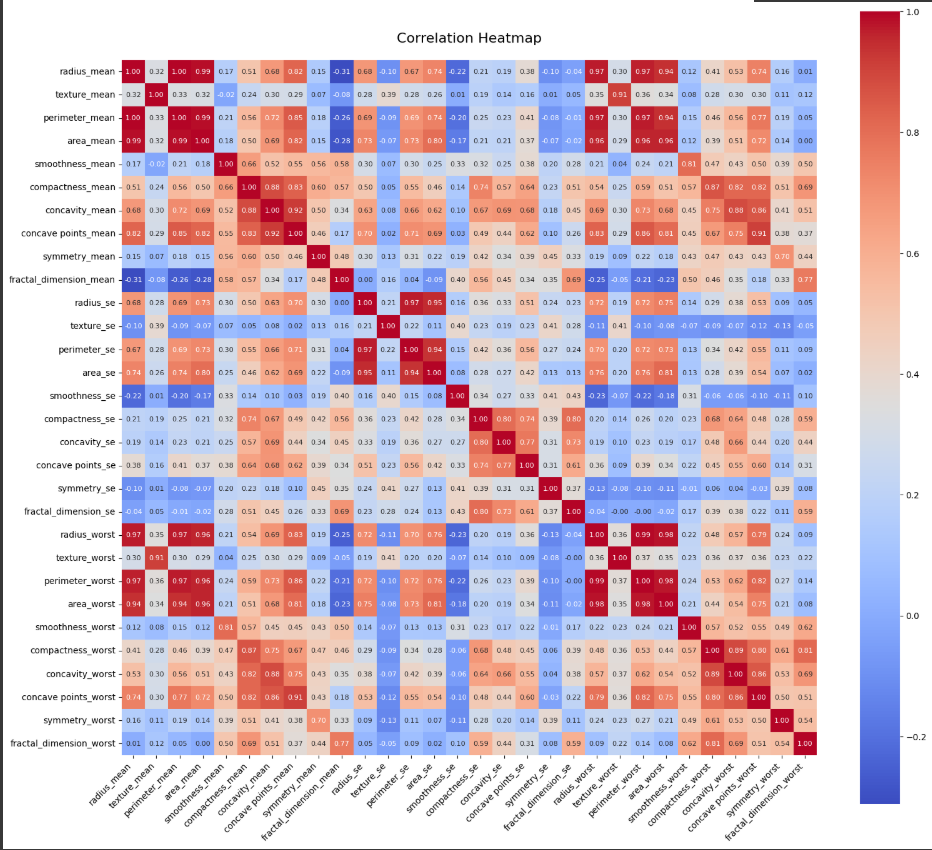
Histogram Output





Correlation Heatmap Output



**6. References / Credits**

* **Pandas Library Documentation**  
  <https://pandas.pydata.org/>
* **Matplotlib Library Documentation**  
  <https://matplotlib.org/>
* **Seaborn Library Documentation**  
  <https://seaborn.pydata.org/>
* **Dataset**: Breast Cancer Wisconsin (Diagnostic) Data Set  
  Source: UCI Machine Learning Repository  
  <https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)>
* **Platform Used**: Google Colab  
  https://colab.research.google.com/