Is Application Development
Big Data
Final Project

Supervised by

Prof. Dr. Taysir Hassan Abdel Hamid

May,2023

Team member

Maha Mahmoud Mohammed (Group 4, Section 2)

Nourhan Mohamed Ali (Group 4, Section 3)

Hala Khaled Mohamed (Group 4, Section 4)

Hend Khaled Lotfy (Group 4, Section 4)

Khloud Farouk Fouad (Group 2, Section 1)

Salma Ahmed Nady (Group 2, Section 3)

Title: “Breast cancer dataset”

Description:

General Description of breast Cancer Dataset:

Breast cancer dataset is a collection of data that contains information about breast cancer patients, their medical history, and various characteristics of the tumor. The dataset typically includes various features such as patient age, tumor size, tumor grade, tumor stage. It may also include information about the patient's family history, previous treatments, and other medical conditions.

What we do in our data set:

In our dataset, we have divided our data into 3 collections which are: test, train, and value. The test collection contains Site ID, Patient ID, Image ID, Lateral View, Presentation, Age, Implant, Device ID, and Predictor.

The train collection contains Site ID, Patient ID, Image ID, Presentation, Age, Cancer, Biopsy, Invasive, Avian, Implantation, Density, Device ID, and Solid Bag.

Val collection contains Site ID, Patient ID, Image ID, lateral view, presentation, age, cancer, biopsy, invasive, BIRADS, implant, density, device id, hard case.

The dataset is available in a tab-delimited text format.

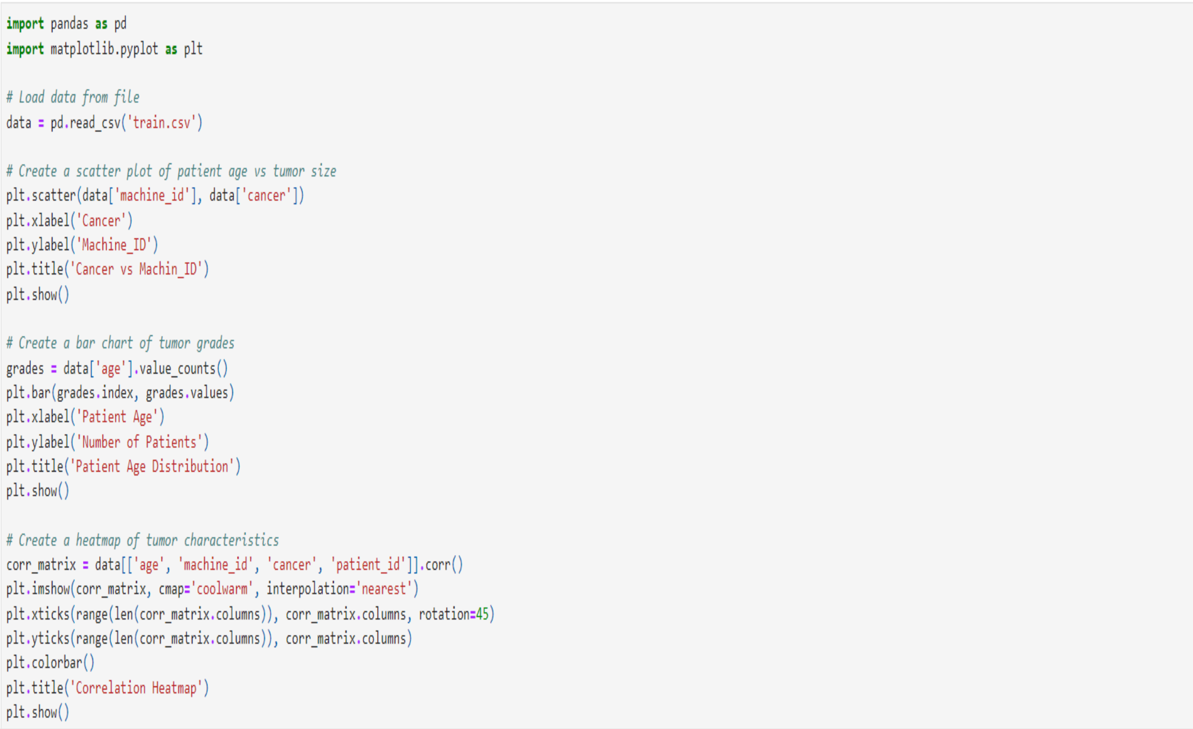
To convert this data set into sets for MongoDB with the relationships between them, we can use three sets: one for train, one for test on this train, and one for value of patient.

We write a Python script to read the data from a tab-delimited text file, convert the data to the above schemas, and insert the documents into the respective MongoDB collections.

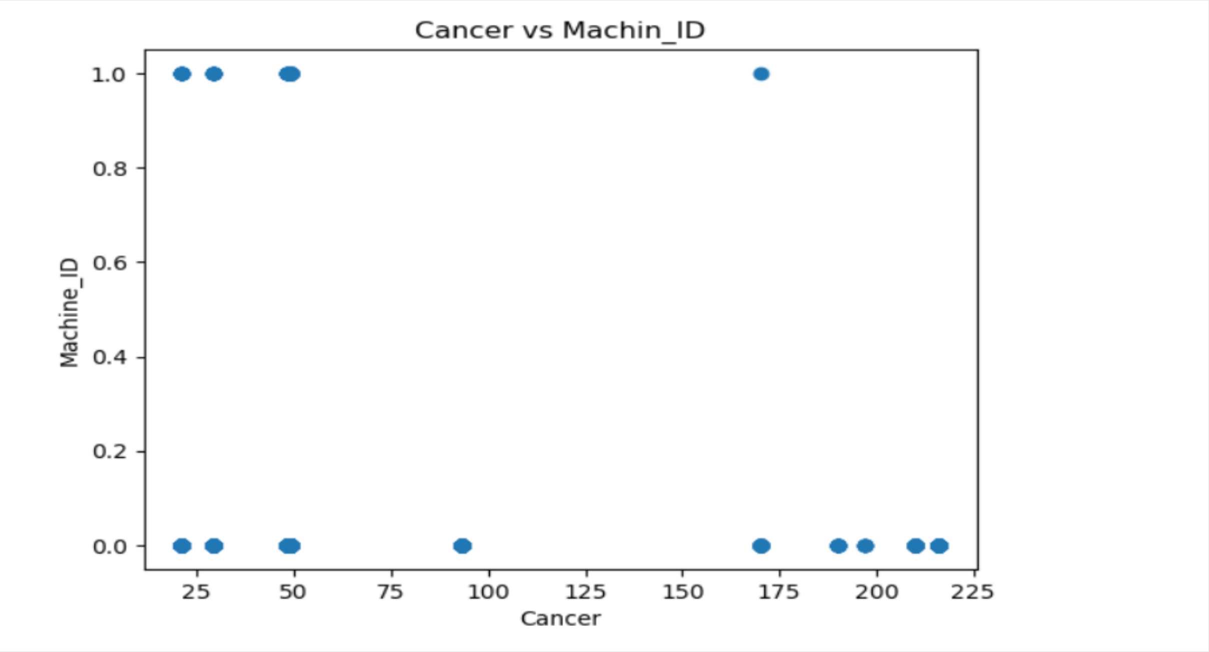
Once the data is in MongoDB, we can perform various queries and analyzes on the data using MongoDB's powerful queries and aggregation capabilities. For example, we can find all data of patient with age a specific number, we can make query on the data, we can insert new data or delete any data you want, we can update the data with new data, we aggregate the data, etc.

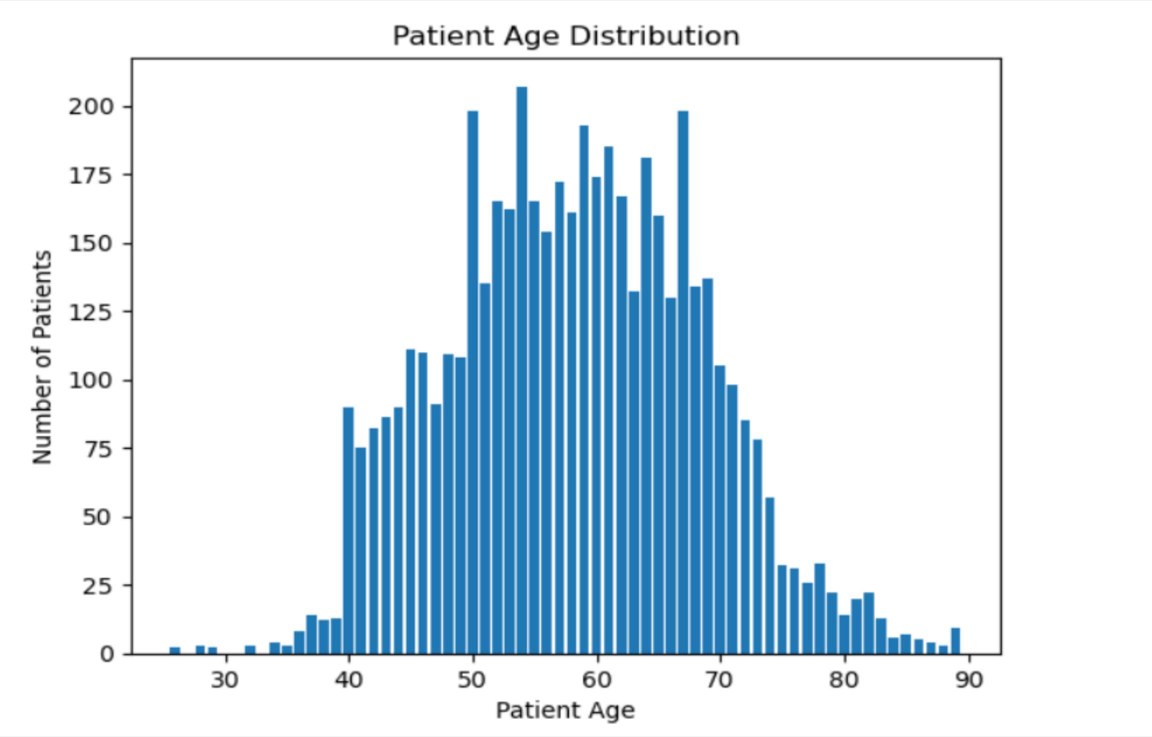
Visualization our data:

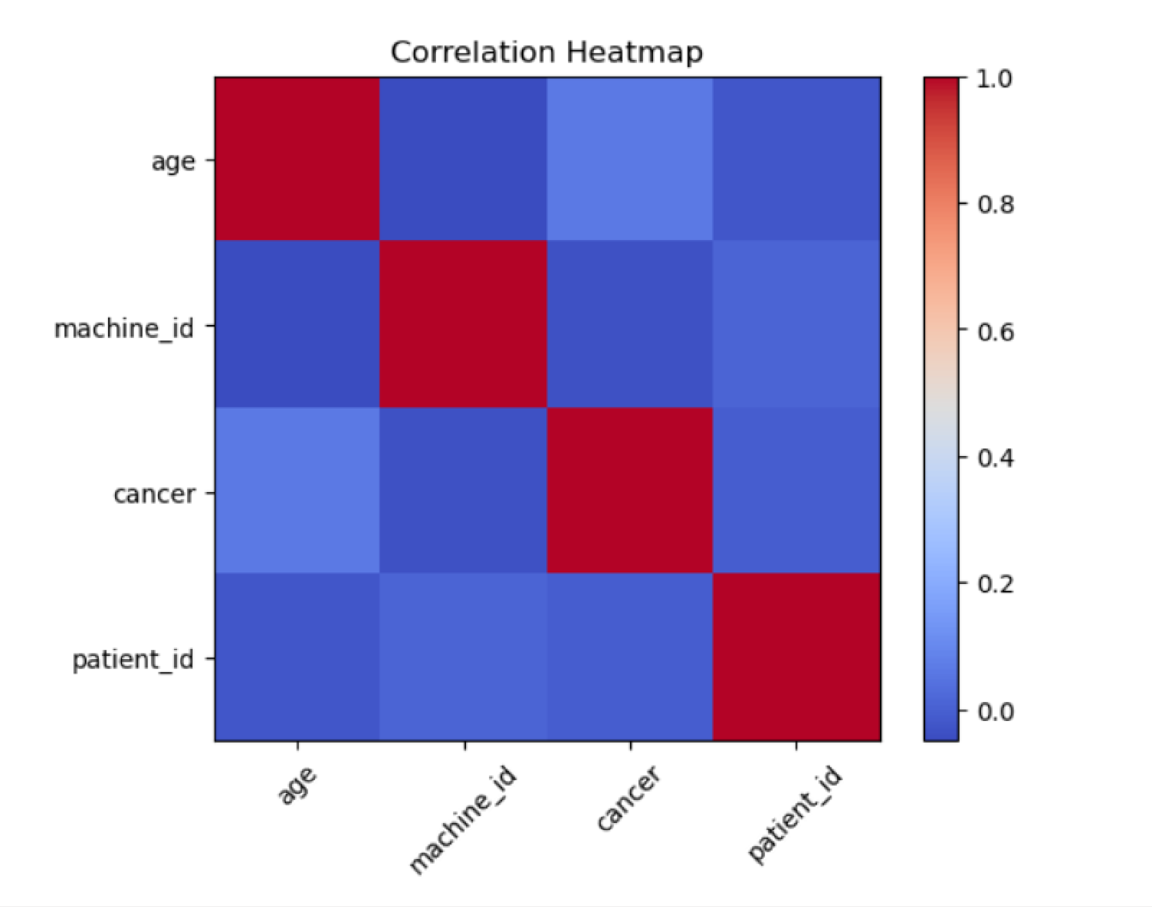
1) Train file:



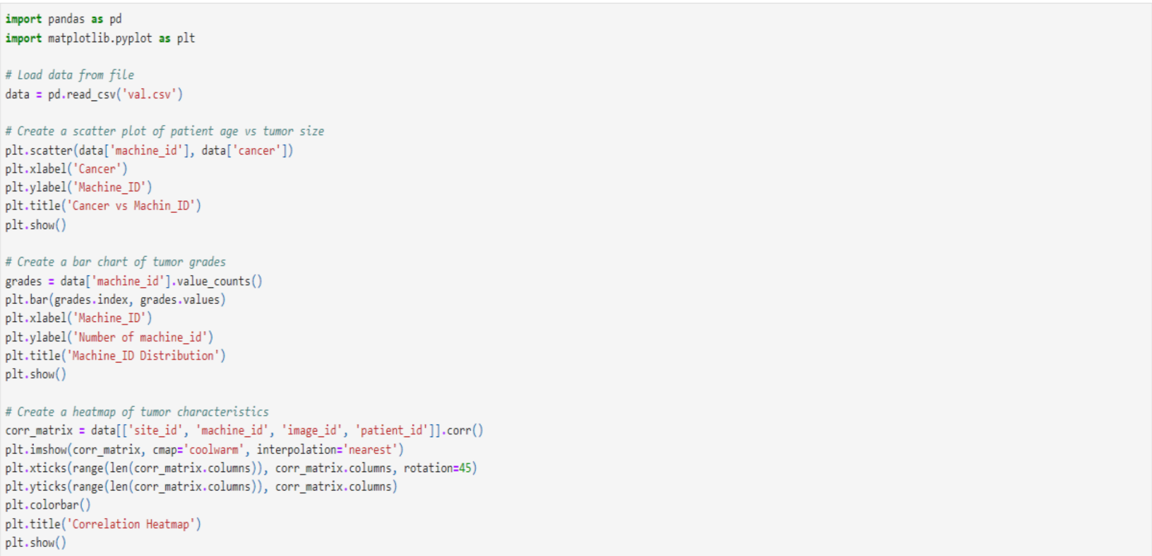
```
import pandas as pd
import matplotlib.pyplot as plt

# Load data from file
data = pd.read_csv('train.csv')

# Create a scatter plot of patient age vs tumor size
plt.scatter(data['machine_id'], data['cancer'])
plt.xlabel('Cancer')
plt.ylabel('Machine_ID')
plt.title('Cancer vs Machin_ID')
plt.show()

# Create a bar chart of tumor grades
grades = data['age'].value_counts()
plt.bar(grades.index, grades.values)
plt.xlabel('Patient Age')
plt.ylabel('Number of Patients')
plt.title('Patient Age Distribution')
plt.show()

# Create a heatmap of tumor characteristics
corr_matrix = data[['age', 'machine_id', 'cancer', 'patient_id']].corr()
plt.imshow(corr_matrix, cmap='coolwarm', interpolation='nearest')
plt.xticks(range(len(corr_matrix.columns)), corr_matrix.columns, rotation=45)
plt.yticks(range(len(corr_matrix.columns)), corr_matrix.columns)
plt.colorbar()
plt.title('Correlation Heatmap')
plt.show()
```



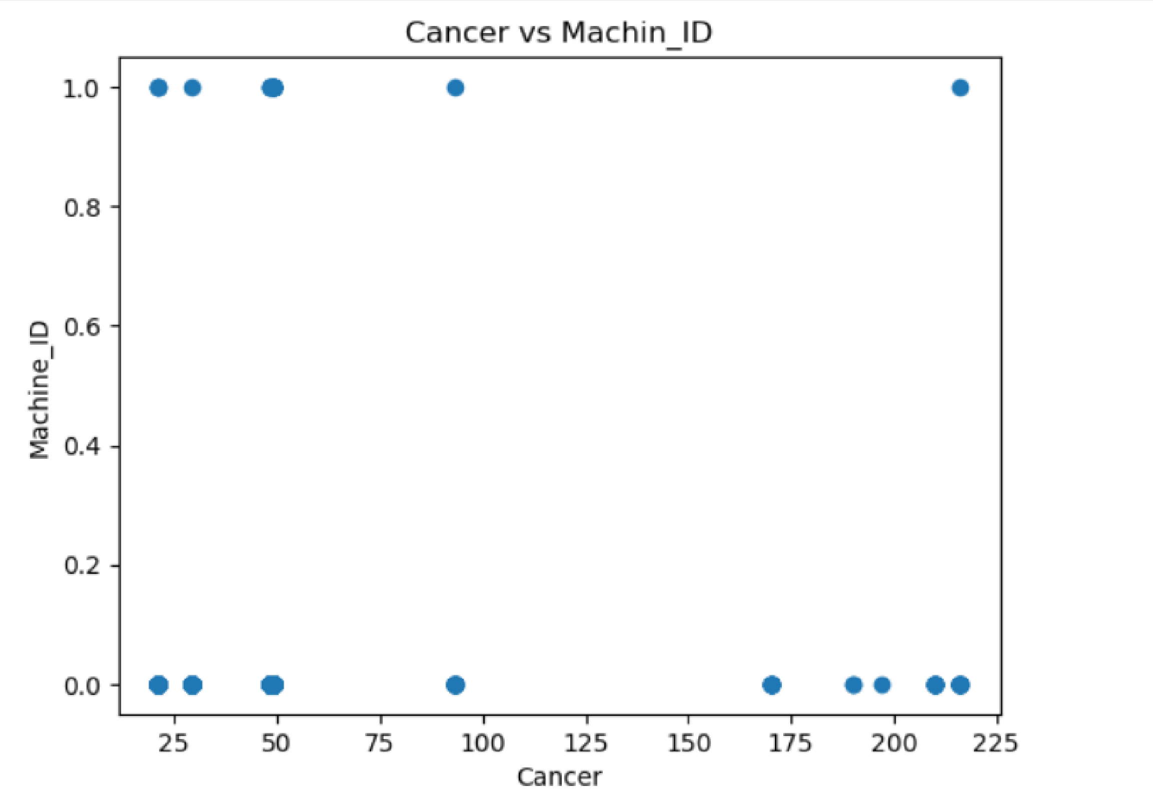

Patient Age Distribution
Number of Patients
0
25
50
75
100
125
150
175
200
30
40
50
60
70
80
90
Patient Age

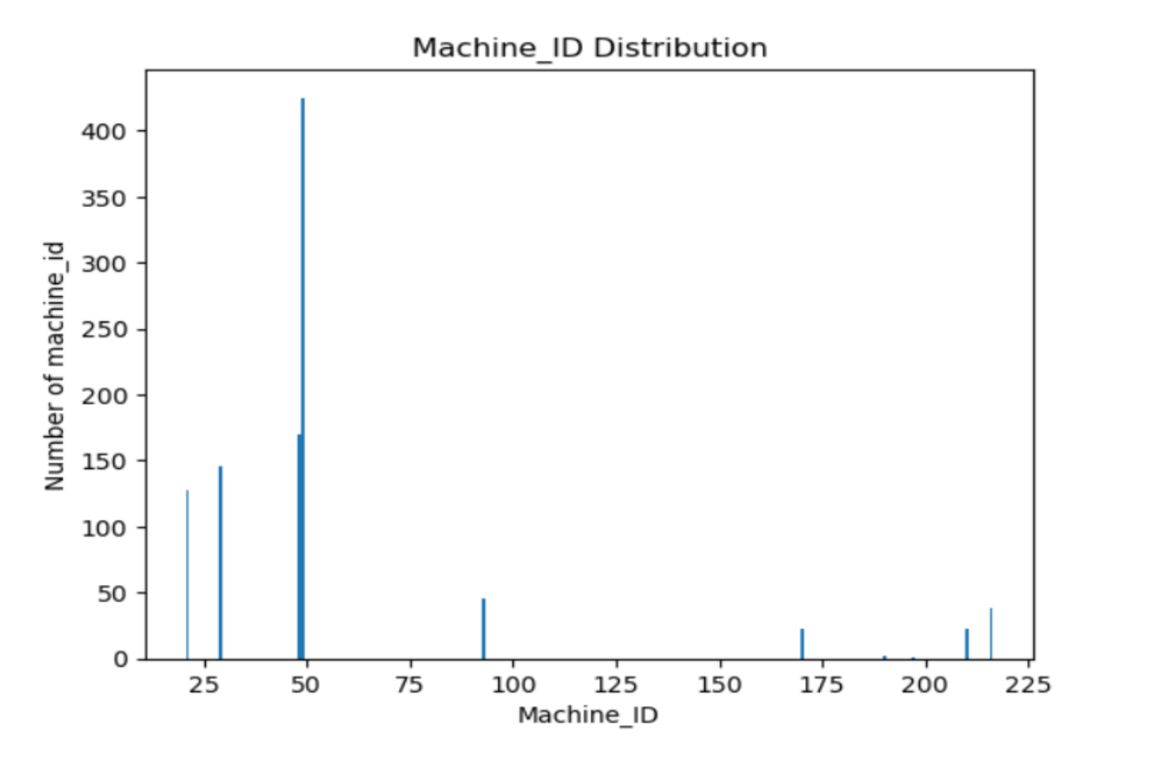

Correlation Heatmap
age
machine_id
cancer
patient_id
age
machine_id
cancer
patient_id
1.0
0.8
0.6
0.4
0.2
0.0

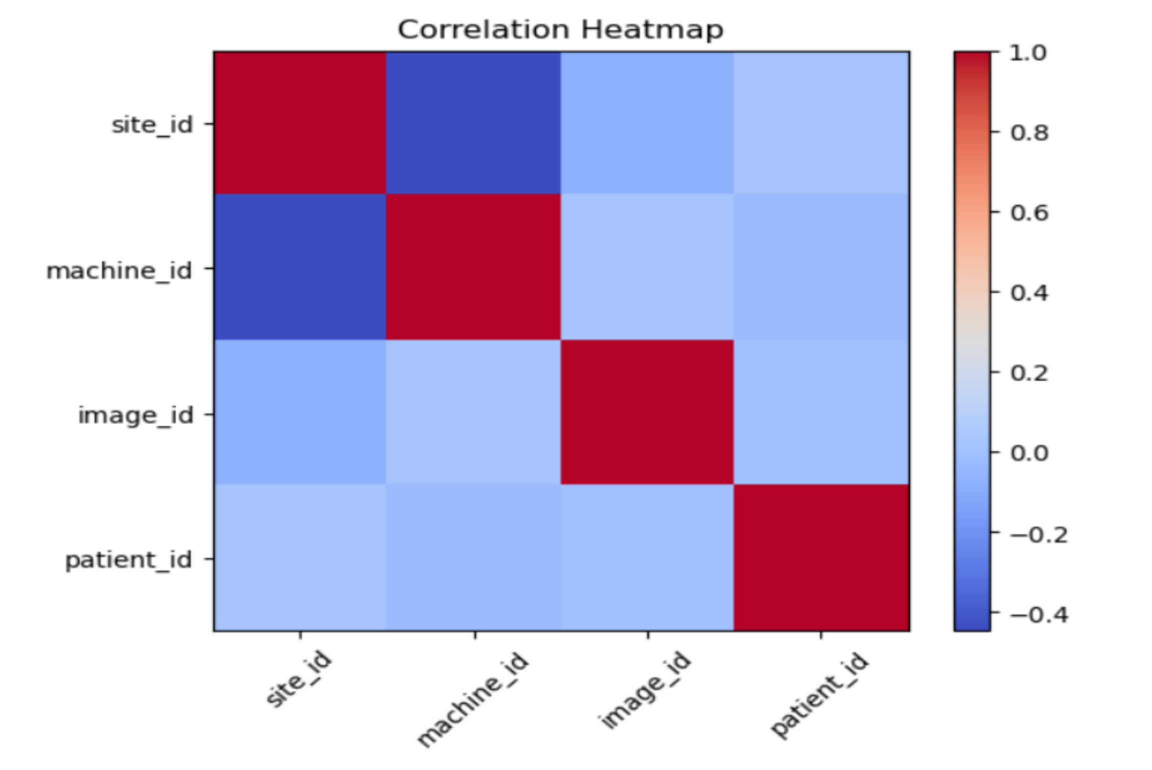
2) Val file:



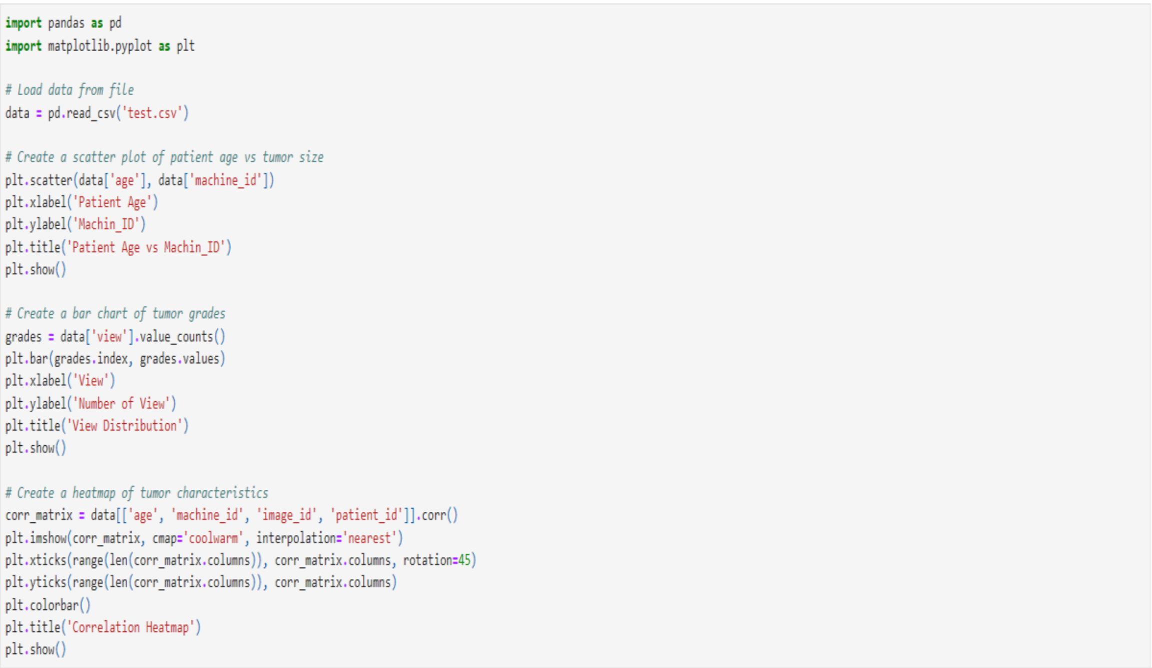
```
import pandas as pd
import matplotlib.pyplot as plt

# Load data from file
data = pd.read_csv('val.csv')

# Create a scatter plot of patient age vs tumor size
plt.scatter(data['machine_id'], data['cancer'])
plt.xlabel('Cancer')
plt.ylabel('Machine_ID')
plt.title('Cancer vs Machin_ID')
plt.show()

# Create a bar chart of tumor grades
grades = data['machine_id'].value_counts()
plt.bar(grades.index, grades.values)
plt.xlabel('Machine_ID')
plt.ylabel('Number of machine_id')
plt.title('Machine_ID Distribution')
plt.show()

# Create a heatmap of tumor characteristics
corr_matrix = data[['site_id', 'machine_id', 'image_id', 'patient_id']].corr()
plt.imshow(corr_matrix, cmap='coolwarm', interpolation='nearest')
plt.xticks(range(len(corr_matrix.columns)), corr_matrix.columns, rotation=45)
plt.yticks(range(len(corr_matrix.columns)), corr_matrix.columns)
plt.colorbar()
plt.title('Correlation Heatmap')
plt.show()
```



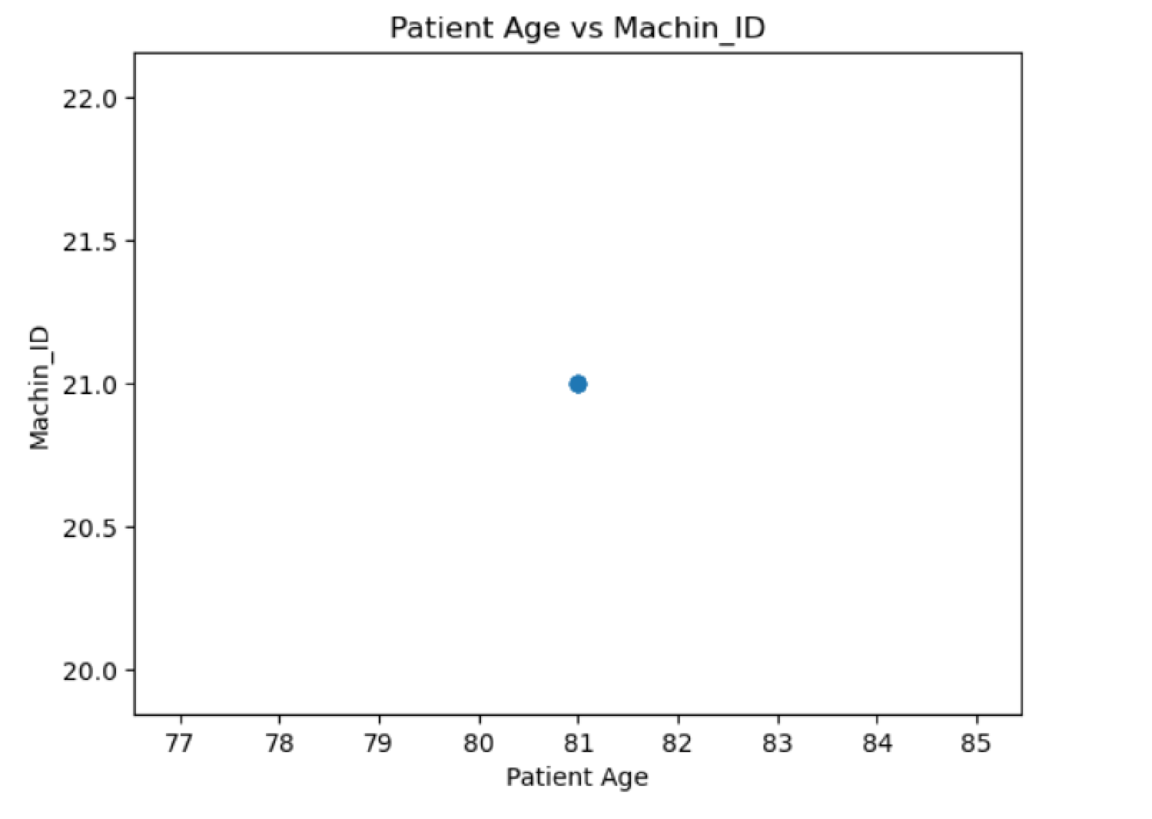

Machine_ID Distribution
Number of machine_id
0
50
100
150
200
250
300
350
400
25
50
75
100
125
150
175
200
225
Machine_ID

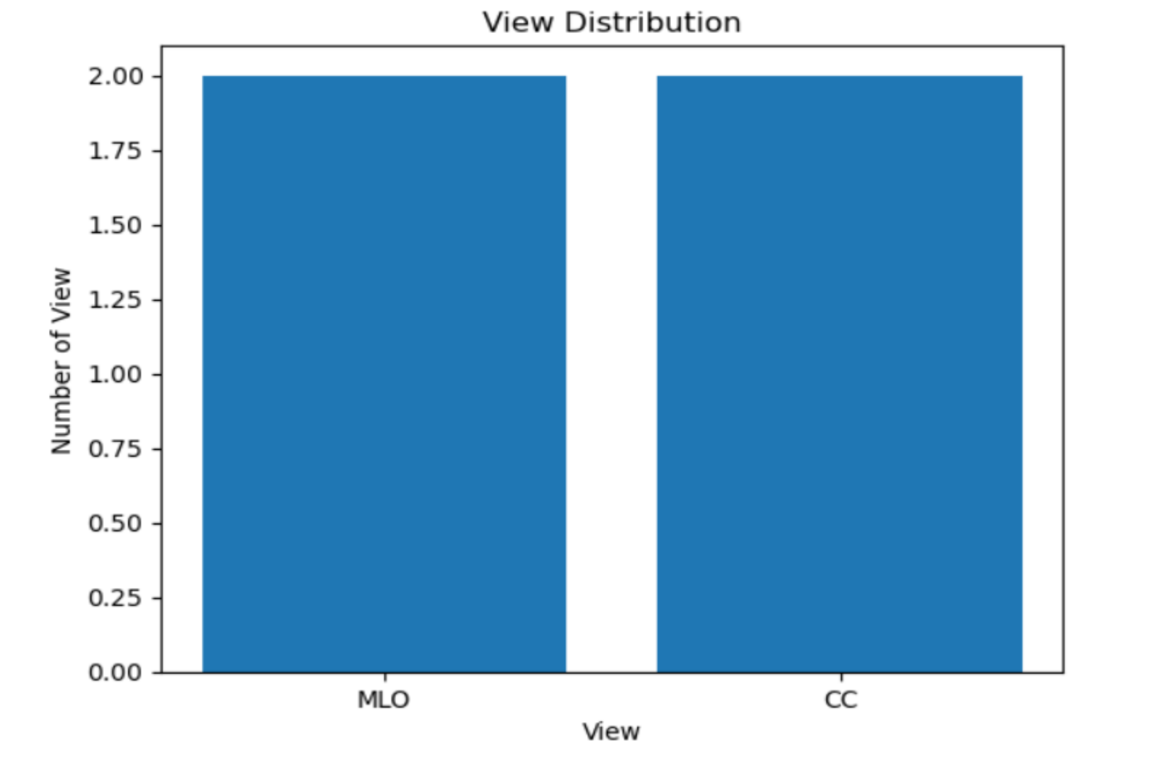

Correlation Heatmap
site_id
machine_id
image_id
patient_id
site_id
machine_id
image_id
patient_id
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

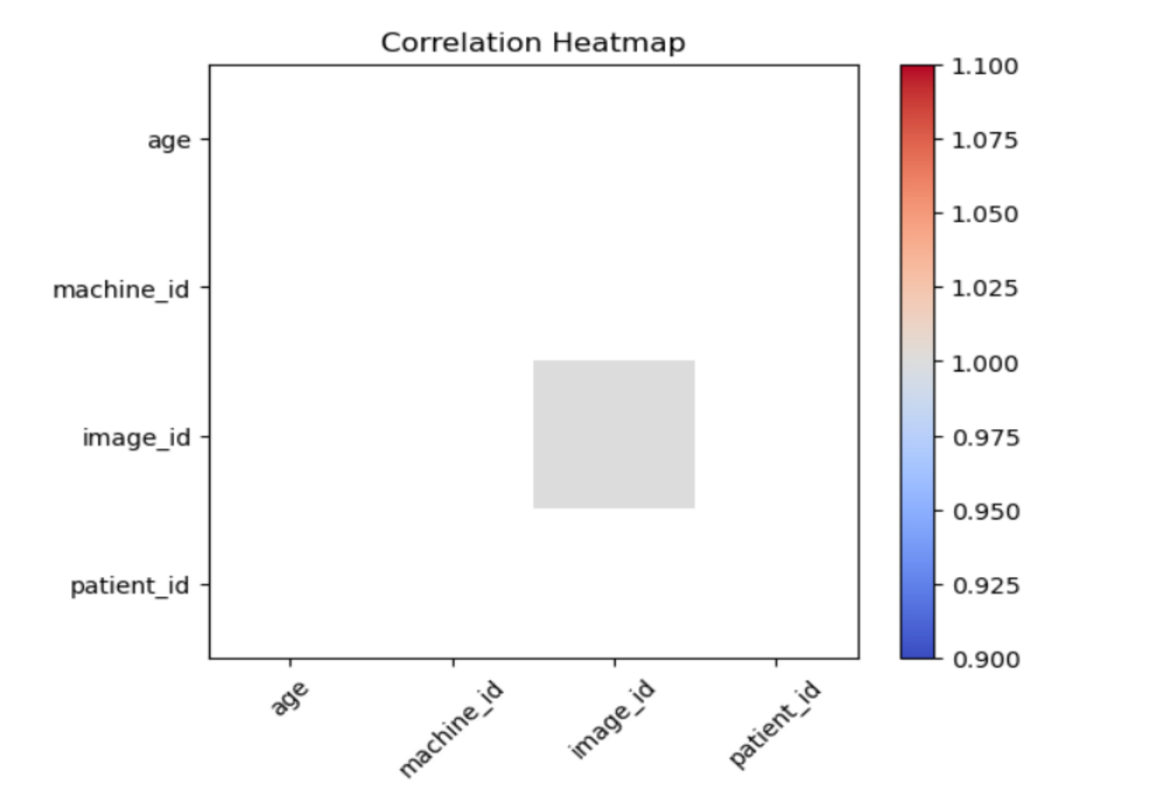
3) Test file:



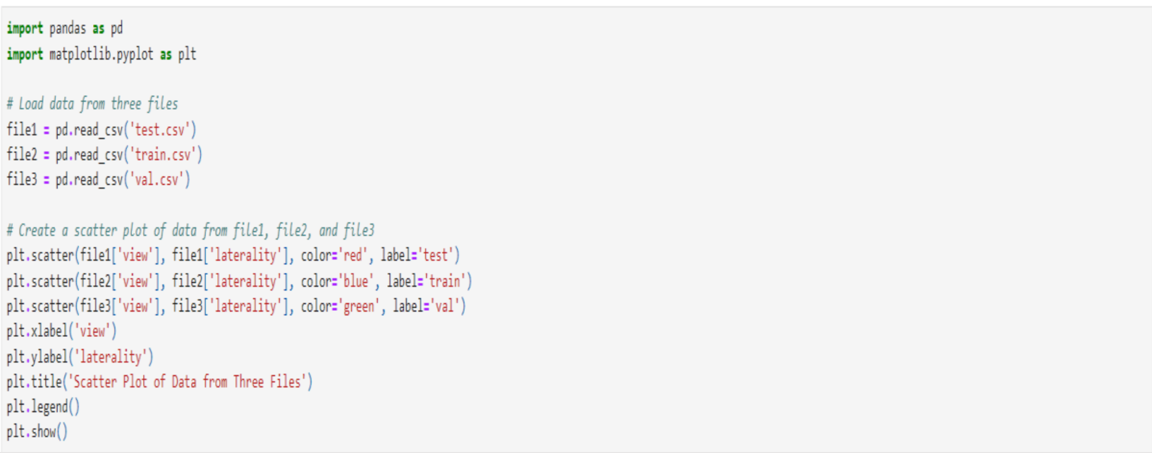
```
import pandas as pd
import matplotlib.pyplot as plt

# Load data from file
data = pd.read_csv('test.csv')

# Create a scatter plot of patient age vs tumor size
plt.scatter(data['age'], data['machine_id'])
plt.xlabel('Patient Age')
plt.ylabel('Machin_ID')
plt.title('Patient Age vs Machin_ID')
plt.show()

# Create a bar chart of tumor grades
grades = data['view'].value_counts()
plt.bar(grades.index, grades.values)
plt.xlabel('View')
plt.ylabel('Number of View')
plt.title('View Distribution')
plt.show()

# Create a heatmap of tumor characteristics
corr_matrix = data[['age', 'machine_id', 'image_id', 'patient_id']].corr()
plt.imshow(corr_matrix, cmap='coolwarm', interpolation='nearest')
plt.xticks(range(len(corr_matrix.columns)), corr_matrix.columns, rotation=45)
plt.yticks(range(len(corr_matrix.columns)), corr_matrix.columns)
plt.colorbar()
plt.title('Correlation Heatmap')
plt.show()
```



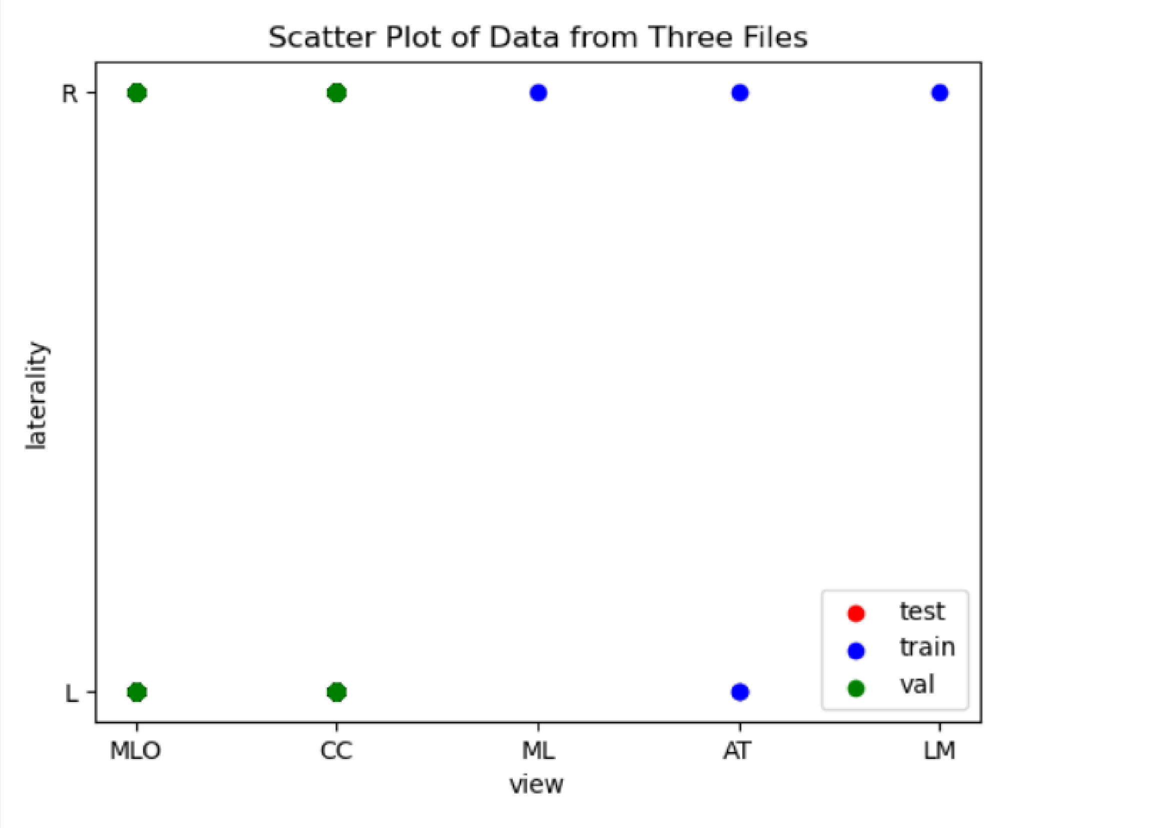

View Distribution
Number of View
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
MLO
CC
View


Correlation Heatmap
age
machine_id
image_id
patient_id
age
machine_id
image_id
patient_id
1.100
1.075
1.050
1.025
1.000
0.975
0.950
0.925
0.900

4) Our 3 collection:



```python
import pandas as pd
import matplotlib.pyplot as plt

# Load data from three files
file1 = pd.read_csv('test.csv')
file2 = pd.read_csv('train.csv')
file3 = pd.read_csv('val.csv')

# Create a scatter plot of data from file1, file2, and file3
plt.scatter(file1['view'], file1['laterality'], color='red', label='test')
plt.scatter(file2['view'], file2['laterality'], color='blue', label='train')
plt.scatter(file3['view'], file3['laterality'], color='green', label='val')
plt.xlabel('view')
plt.ylabel('laterality')
plt.title('Scatter Plot of Data from Three Files')
plt.legend()
plt.show()
```


Scatter Plot of Data from Three Files
laterality
R
L
view
MLO
CC
ML
AT
LM
● test
● train
● val