

Data Collection and Preprocessing Phase

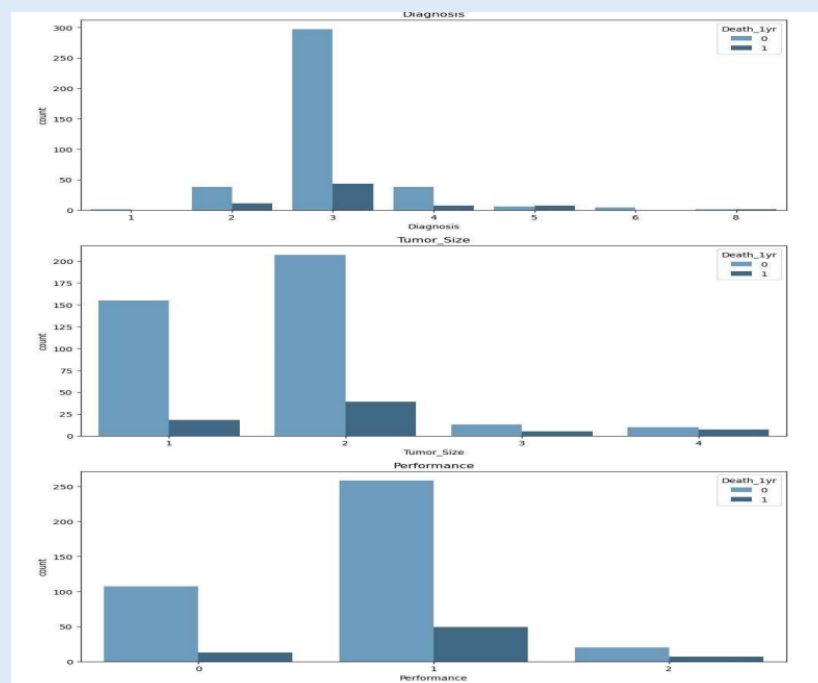
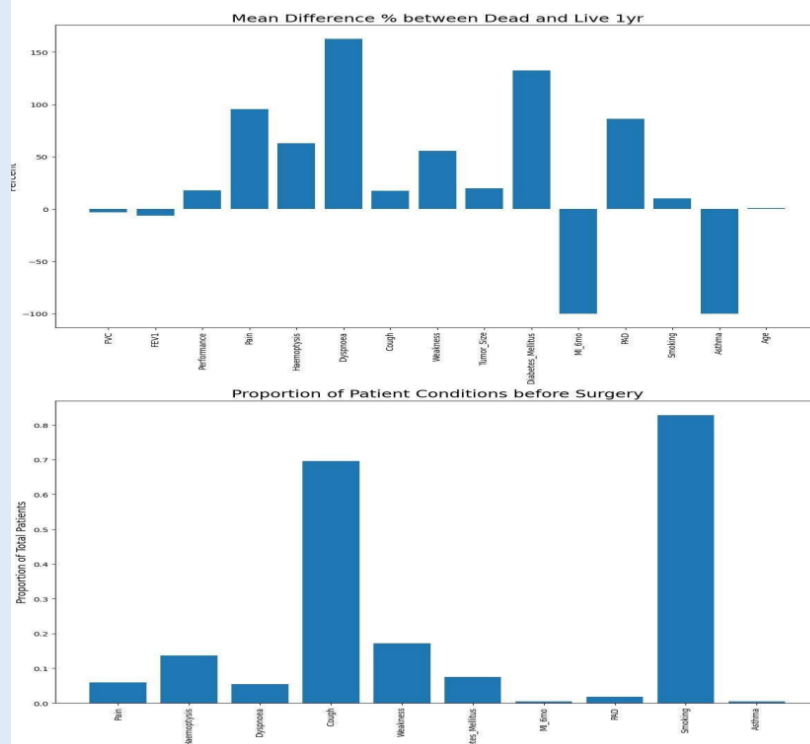
Date	24 April 2024
Team ID	739847
Project Title	One Year Life Expectancy post on Thoracic Surgery using Machine Learning
Maximum Marks	6 Marks

Data Exploration and Preprocessing Report

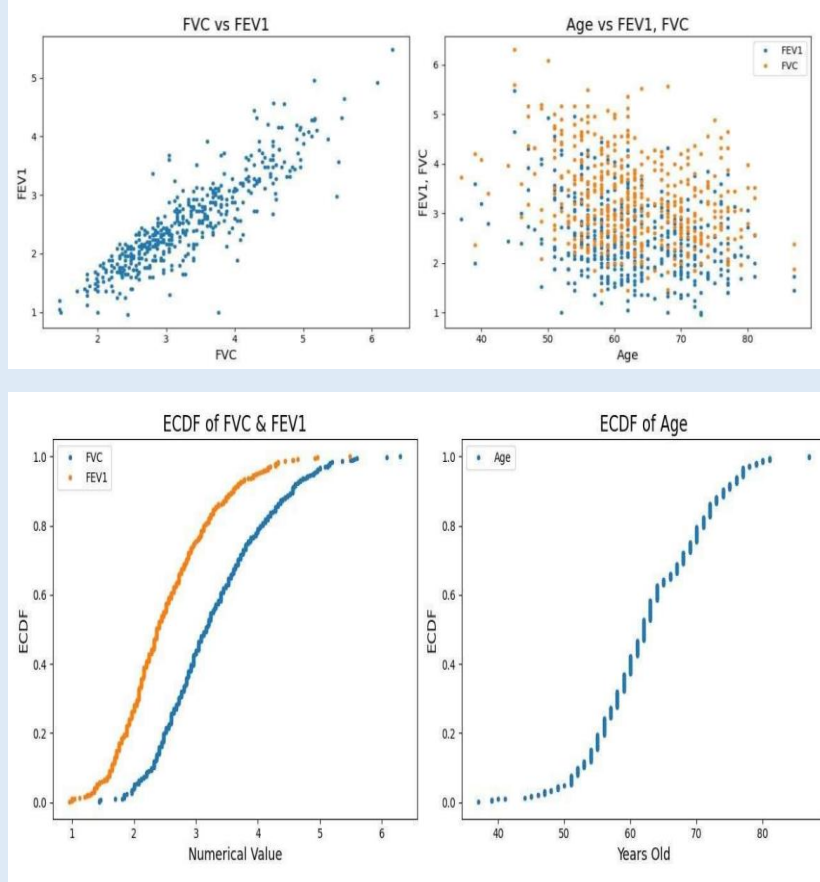
Dataset variables will be statistically analyzed to identify patterns and outliers, with Python employed for preprocessing tasks like normalization and feature engineering. Data cleaning will address missing values and outliers, ensuring quality for subsequent analysis and modeling, and forming a strong foundation for insights and predictions.

Section	Description
Data Overview	<u>Dimension:</u> 454 rows × 17 columns
	<u>Descriptive statistics:</u>

Exploratory Data analysis



Correlation Coefficient



Data Preprocessing Code Screenshots

Loading Data

```
In [29]: # Import necessary libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, f1_score, classification_report, confusion_matrix
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
import itertools
import warnings

# Ignore warnings
warnings.filterwarnings('ignore')

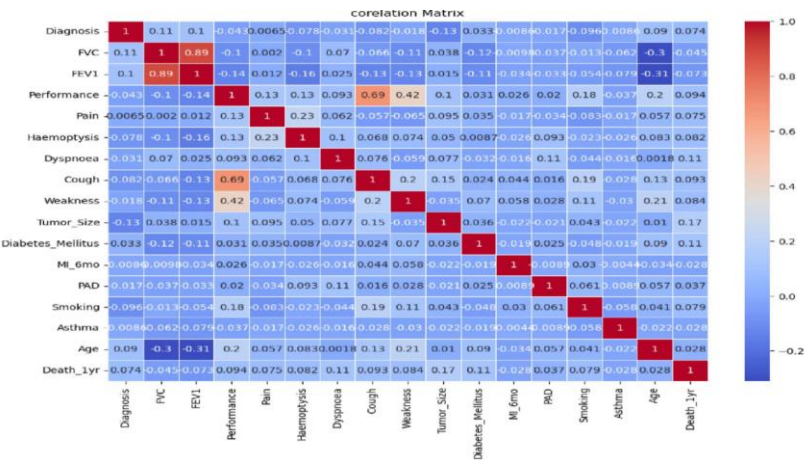
# Load your dataset
df = pd.read_csv('ThoracicSurgery.csv')

# Feature selection
# Select features relevant for prediction
features = ['FVC', 'FEV1', 'Performance', 'Pain', 'Haemoptysis', 'Dyspnoea',
           'Cough', 'Weakness', 'Tumor_Size', 'Diabetes_Mellitus', 'MI_6mo',
           'PAD', 'Smoking', 'Asthma', 'Age']
target = 'Death_1yr'

# Prepare the data
X = df[features]
y = df[target]

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Correlation Matrix



Data Transformation

```
In [24]: x=df.iloc[:,0:15].values
         y=df.iloc[:,15:16].values

In [25]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)

In [26]: print('Shape of x_train {}'.format(x_train.shape))
         print('Shape of y_train {}'.format(y_train.shape))
         print('Shape of x_test {}'.format(x_test.shape))
         print('Shape of y_test {}'.format(y_test.shape))

         Shape of x_train (363, 15)
         Shape of y_train (363, 1)
         Shape of x_test (91, 15)
         Shape of y_test (91, 1)

In [27]: from sklearn.preprocessing import StandardScaler

         # Standard scaling
         sc = StandardScaler()
         x_train = sc.fit_transform(x_train)
         x_test = sc.transform(x_test)
```

Feature Engineering

Attached the codes in final submission.

Save Processed Data

Data saved in the form of model .pkl file

