AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Assignment Title:	Predicting Cancer Using Data Analysis and Machine Learning Approach			
Assignment No:	01		Date of Submission:	8 May 2023
Course Title:	Machine Learning			
Course Code:	01628		Section:	A
Semester:	Spring	2022-23	Course Teacher:	Prof. Dr. Md. Asraf Ali

Declaration and Statement of Authorship:

1. I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student's work or from any other source except where due acknowledgement is made.
3. No part of this Assignment/Case-Study has been written for me/us by any other person except where such collaborationhas been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the Faculty for review and comparison, including review by external examiners.
7. I/we understand thatPlagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a formofcheatingandisaveryseriousacademicoffencethatmayleadtoexpulsionfromtheUniversity. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, including electronic data, and oral presentations. Plagiarism occurs when the origin of them arterial used is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy my/our work.

* *Student(s) must complete all details except the faculty use part.*
** Please submit all assignments to your course teacher or the office of the concerned teacher.

Group Name/No.: 02

No	Name	ID	Program	Signature
1	Arpita Saha	19-41363-3	BSc CSE	
2	MD. Shahariar Rashid	19-41372-3	BSc CSE	
3	Jakia Sultana Nupur	19-41405-3	BSc CSE	
4	Tanzim Zaman Sany	19-41409-3	BSc CSE	

Faculty use only		
FACULTYCOMMENTS	**Marks Obtained**	
	Total Marks	

Predicting Cancer Using Data Analysis and Machine Learning Approach

1. Project Objective

The objective of this project is to analyze the dataset of cancer patients' lifestyles and identify patterns that can potentially contribute to the development or prevention of cancer. By using big data analysis techniques, this project aims to provide valuable insights that can aid in the fight against cancer and help reduce the number of premature deaths caused by this disease. Specifically, the project will focus on identifying lifestyle factors such as diet, exercise, and smoking habits that may impact the development of cancer, as well as any correlations between these factors and specific types of cancer. The ultimate goal of this project is to improve our understanding of cancer and develop strategies to prevent and treat the disease.

2. Project Methodology

2.1 Data Collection Procedure

The dataset, named Cancer Patients Data, that was used in this project was been collected from Kaggle which was updated by Rishi Damarla. Raw data was collected from Data World and the author was Prithivraj, where the dataset is named as Lung cancer data. The dataset has about 1000 instances and 25 attributes. The data were collected from numerous cancer patients. It has a categorical attribute which is called Level and the unique attribute is Patient ID. All other attributes are numerical. The dataset reflects the several health conditions and diverse habits of the patients.

2.2 Data Validation Procedure

The dataset was licensed under Creative Commons organization (CC0 1.0). The owner of this dataset has devoted the work to the public domain by making the access free of the dataset. Anyone can use the dataset both for commercial and non-commercial usages without asking for any permission.

2.3 Data Preprocessing and Normalization

We had to preprocess the dataset and normalize if its needed. By doing all these the accuracy level of machine learning approach will be higher than before.

1. We checked the columns and looked for null values.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 25 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Patient Id                1000 non-null   object
 1   Age                       1000 non-null   int64
 2   Gender                    1000 non-null   int64
 3   Air Pollution             1000 non-null   int64
 4   Alcohol use               1000 non-null   int64
 5   Dust Allergy              1000 non-null   int64
 6   OccuPational Hazards      1000 non-null   int64
 7   Genetic Risk              1000 non-null   int64
 8   chronic Lung Disease      1000 non-null   int64
 9   Balanced Diet             1000 non-null   int64
 10  Obesity                   1000 non-null   int64
 11  Smoking                   1000 non-null   int64
 12  Passive Smoker            1000 non-null   int64
 13  Chest Pain                1000 non-null   int64
 14  Coughing of Blood         1000 non-null   int64
 15  Fatigue                   1000 non-null   int64
 16  Weight Loss               1000 non-null   int64
 17  Shortness of Breath       1000 non-null   int64
 18  Wheezing                  1000 non-null   int64
 19  Swallowing Difficulty     1000 non-null   int64
 20  Clubbing of Finger Nails  1000 non-null   int64
 21  Frequent Cold             1000 non-null   int64
 22  Dry Cough                 1000 non-null   int64
 23  Snoring                   1000 non-null   int64
 24  Level                     1000 non-null   object
dtypes: int64(23), object(2)
memory usage: 195.4+ KB

No null value found so data is clean
```

Fig 01: Result of finding empty values in the data frame

2. Check the std, min, max, mean and percentile values of the dataframe.

```
cancer_patient.describe().T
```

	count	mean	std	min	25%	50%	75%	max
Age	1000.0	37.174	12.005493	14.0	27.75	36.0	45.0	73.0
Gender	1000.0	1.402	0.490547	1.0	1.00	1.0	2.0	2.0
Air Pollution	1000.0	3.840	2.030400	1.0	2.00	3.0	6.0	8.0
Alcohol use	1000.0	4.563	2.620477	1.0	2.00	5.0	7.0	8.0
Dust Allergy	1000.0	5.165	1.980833	1.0	4.00	6.0	7.0	8.0
OccuPational Hazards	1000.0	4.840	2.107805	1.0	3.00	5.0	7.0	8.0
Genetic Risk	1000.0	4.580	2.126999	1.0	2.00	5.0	7.0	7.0
chronic Lung Disease	1000.0	4.380	1.848518	1.0	3.00	4.0	6.0	7.0
Balanced Diet	1000.0	4.491	2.135528	1.0	2.00	4.0	7.0	7.0
Obesity	1000.0	4.465	2.124921	1.0	3.00	4.0	7.0	7.0
Smoking	1000.0	3.948	2.495902	1.0	2.00	3.0	7.0	8.0
Passive Smoker	1000.0	4.195	2.311778	1.0	2.00	4.0	7.0	8.0
Chest Pain	1000.0	4.438	2.280209	1.0	2.00	4.0	7.0	9.0
Coughing of Blood	1000.0	4.859	2.427965	1.0	3.00	4.0	7.0	9.0
Fatigue	1000.0	3.856	2.244616	1.0	2.00	3.0	5.0	9.0
Weight Loss	1000.0	3.855	2.206546	1.0	2.00	3.0	6.0	8.0
Shortness of Breath	1000.0	4.240	2.285087	1.0	2.00	4.0	6.0	9.0
Wheezing	1000.0	3.777	2.041921	1.0	2.00	4.0	5.0	8.0
Swallowing Difficulty	1000.0	3.746	2.270383	1.0	2.00	4.0	5.0	8.0
Clubbing of Finger Nails	1000.0	3.923	2.388048	1.0	2.00	4.0	5.0	9.0
Frequent Cold	1000.0	3.536	1.832502	1.0	2.00	3.0	5.0	7.0
Dry Cough	1000.0	3.853	2.039007	1.0	2.00	4.0	6.0	7.0
Snoring	1000.0	2.926	1.474686	1.0	2.00	3.0	4.0	7.0

Fig 02: Result for instances count, mean, max, min, std and percentile values

3. We removed irrelevant column (Patient Id)

	Age	Gender	Air Pollution	Alcohol use	Dust Allergy	OccuPational Hazards	Genetic Risk	chronic Lung Disease	Balanced Diet	Obesity	Smoking	Passive Smoker	Chest Pain	Coughing of Blood	Fatigue	Weight Loss	Shortn of Bre
0	33	1	2	4	5	4	3	2	2	4	3	2	2	4	3	4	
1	17	1	3	1	5	3	4	2	2	2	2	4	2	3	1	3	
2	35	1	4	5	6	5	5	4	6	7	2	3	4	8	8	7	
3	37	1	7	7	7	7	6	7	7	7	7	7	7	8	4	2	
4	46	1	6	8	7	7	7	6	7	7	8	7	7	9	3	2	

Fig 03: Result after removing Patient Id attribute

4. Converted the categorical data into numerical data (Level column)

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 24 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Age                       1000 non-null   int64
 1   Gender                    1000 non-null   int64
 2   Air Pollution             1000 non-null   int64
 3   Alcohol use               1000 non-null   int64
 4   Dust Allergy              1000 non-null   int64
 5   OccuPational Hazards      1000 non-null   int64
 6   Genetic Risk              1000 non-null   int64
 7   chronic Lung Disease      1000 non-null   int64
 8   Balanced Diet             1000 non-null   int64
 9   Obesity                   1000 non-null   int64
 10  Smoking                   1000 non-null   int64
 11  Passive Smoker            1000 non-null   int64
 12  Chest Pain                1000 non-null   int64
 13  Coughing of Blood         1000 non-null   int64
 14  Fatigue                   1000 non-null   int64
 15  Weight Loss               1000 non-null   int64
 16  Shortness of Breath       1000 non-null   int64
 17  Wheezing                  1000 non-null   int64
 18  Swallowing Difficulty     1000 non-null   int64
 19  Clubbing of Finger Nails  1000 non-null   int64
 20  Frequent Cold             1000 non-null   int64
 21  Dry Cough                 1000 non-null   int64
 22  Snoring                   1000 non-null   int64
 23  Level                     1000 non-null   int64
dtypes: int64(24)
memory usage: 187.6 KB
```

Fig 04: Result after converting Level attribute into numeric

5. Normalize dataset.

```
array([[-0.89787225, -0.20867982, -0.07875426, ..., -1.06222082,
        -0.351619  , -0.37899803],
       [-0.40677085, -1.35007771, -0.07875426, ..., -1.06222082,
        -0.76163409, -1.26601468],
       [ 0.08433054,  0.17178615,  0.4242568 , ..., -0.18880005,
         1.28844135,  1.83854361],
       ...,
       [ 0.08433054,  0.17178615,  0.4242568 , ..., -0.18880005,
         1.28844135,  1.83854361],
       [ 1.06653334,  1.31318404,  0.92726785, ...,  1.12133111,
         1.69845643, -0.37899803],
       [ 1.06653334,  0.17178615,  0.4242568 , ..., -0.18880005,
         1.28844135,  1.83854361]])
```

Fig 05: Result after normalize the dataset

2.4 Feature extraction procedure

1. View outliers

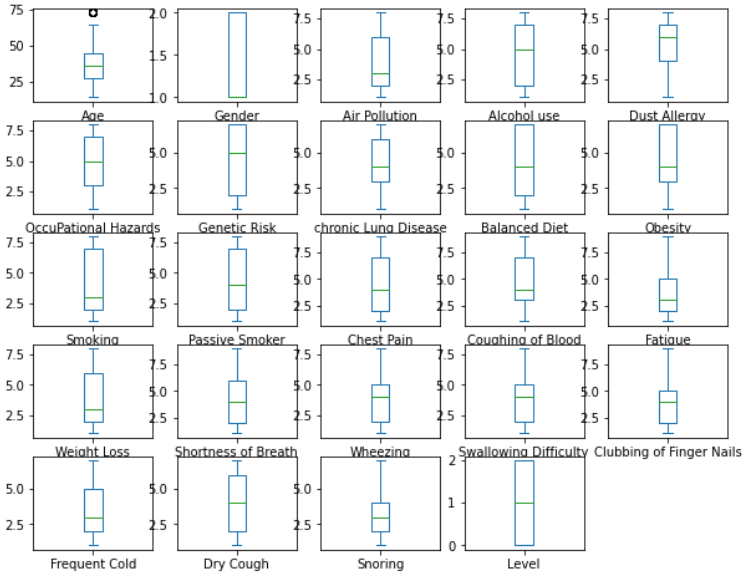


Fig 06: Outliers

2. View histogram

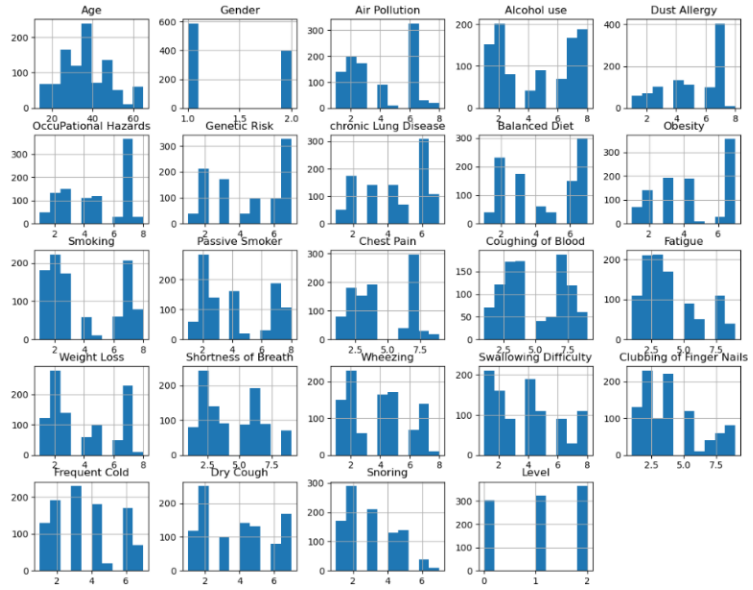


Fig 07: Histogram

3. View correlation graph (Heat map)

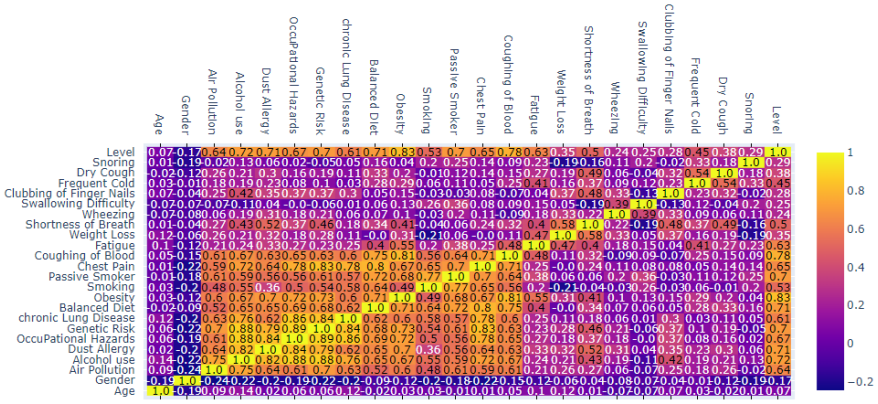
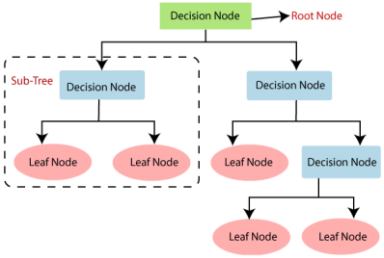


Fig 08: Correlation Graph

2.5 Classification Algorithms

Classification algorithms in machine learning use input training data to predict the likelihood that subsequent data will fall into one of the predetermined categories. There are main five types of classification algorithms in machine learning. They are – Logistic Regression, Naïve Bayes, KNN, Decision Tree and SVM. We implemented two types of algorithms: Decision Tree algorithm and Cross-Validation process.

- Decision Tree: Decision technique that can be Regression problems, solving Classification classifier, where features of a dataset, rules and each leaf



Tree is a Supervised learning used for both classification and but mostly it is preferred for problems. It is a tree-structured internal nodes represent the branches represent the decision node represents the outcome.

- Cross Validation: Cross-validation is a technique for evaluating ML models by training several ML models on subsets of the available input data and evaluating them on the complementary subset of the data.

2.6 Data Analysis Techniques

Machine Learning in data analytics involves the use of techniques such as clustering, elasticity, and natural language. In clustering, it is for the machine to decide the commonalities between the different datasets to understand how certain things are alike. Linear regression algorithms look for correlations between continuous variables innately. On the other hand, logistic regression is used for classifying categorical data. It is yet another technique borrowed from the field of statistics. A Logistic Regression can be referred to as a Linear Regression model but the former uses a complex cost function which is called the 'Sigmoid function' or 'logistic function' instead of a linear function. The sigmoid function plots any real value into an alternate value in the range 0 to 1. In machine learning, the sigmoid (the S-shaped curve) is employed to map projections to probabilities. Multi-class classification is also supported with logistic regression by using one v/s rest scheme. In the one v/s all method, while working with one class at a time, that class is denoted by 1 and the remaining by 0 and their results are combined to get the final fit. The Decision-tree model falls under the supervised learning category. But unlike other supervised learning algorithms, this particular algorithm can even be used for solving regression and classification problems. It is largely used to help decide about any process. This model is basically a rule-based approach where a tree-like structure is

created. Learning starts from the top of the tree (i.e. the root node). Each node basically consists of a question, to which the answer is positive or negative. The questions at different levels are related to the different attributes in the dataset. Based on the answers at different levels of the tree, the algorithm concludes as to what should be the output correspond to the input sample. It is a very popular algorithm, mainly due to its simplicity. The benefit of this algorithm is that for some input samples, it can predict the output quickly, without even traversing a major portion of the tree. But that depends entirely on the dataset. Depending on the kind of target variables, Decision-trees come in two types - Categorical Variable Decision Tree and Continuous Variable Decision Tree. Machine Learning is used in Data Analytics to decipher patterns, understand customer behavior and segmentation, help in decision-making, etc.

2.7 Block Diagram and Workflow of Proposed Model

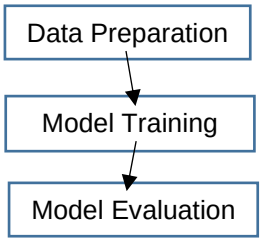


Fig 09: Block Diagram

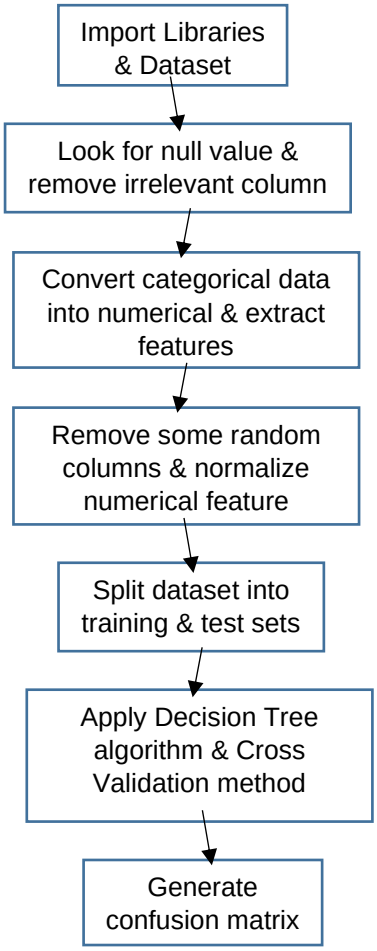


Fig 09: Workflow Diagram

2.8 Experimental Setup and Implementations

- Install and import all the necessary libraries
- Import the dataset and check the instances and attributes
- Check for null value and remove irrelevant column
- Convert the categorical instance into numerical
- Plot boxplot to view outliers
- Capture the outlier using IQR for the unique value
- Remove outlier for unique value
- Plot histogram to check data distribution
- Plot correlation graph
- Select the highly correlated columns with targeted variable
- Drop some other columns and creating feature dataframe(x) and lable dataframe(y)
- Normalize numerical features
- Split data into train & test sets
- Apply Cross Validation process and Decision Tree algorithm
- Generate confusion matrix and print classification report for both

3. Results and Discussion

3.1 Results Comparison

After implementing Cross Validation process using greed search method achieved the accuracy of 93.43%. Finally, Decision Tree algorithm has achieved the accuracy of 100% for the dataset. Here we can be assuring that Decision Tree algorithm works the best for this dataframe.

3.2 Confusion Matrix Analysis

The accuracy of confusion matrix generated by Logistic Regression method is 94%. Cross Validation process's confusion matrix's accuracy is 93%. Accuracy achieved by confusion matrix which was generated by Decision Tree algorithm is 100%. Here we can tell Decision Tree algorithm ensures the highest accuracy rate for the dataset.

A confusion matrix is a table that will categorize the predictions against the actual values. It includes two dimensions among them one will indicate the predicted values and another one will represent the actual values.

For Cross Validation confusion matrix, the positive class is "High", and the negative classes are "Medium" and "Low".

True Positive (TP): Positive values are correctly predicted. Here, TP value is 50. That means these 50 instances are correctly classified into High class.

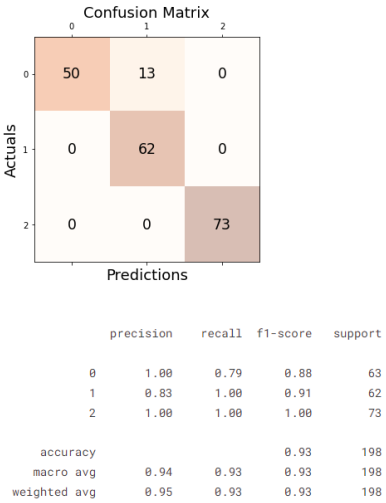
False Positive (FP): Negative values are incorrectly predicted as positive. Here FP value is 0.

False Negative (FN): Positive values predicted as negative. In this confusion matrix the FN value is 13

True Negative (TN): Negative values predicted as actual negative values. Here, TN values are 62, 73.

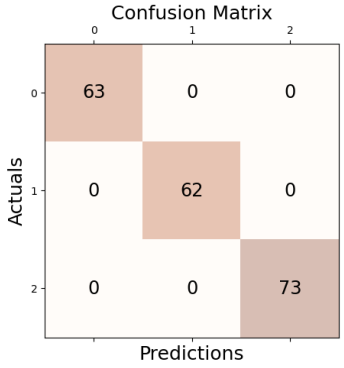
For Decision Tree Confusion matrix here, our accuracy is 100%. That means, TP rate and TN rate is 100% and FP rate and FN rate is 0%.

3.3 Graphical Representation of Results



	precision	recall	f1-score	support
0	1.00	0.79	0.88	63
1	0.83	1.00	0.91	62
2	1.00	1.00	1.00	73
accuracy			0.93	198
macro avg	0.94	0.93	0.93	198
weighted avg	0.95	0.93	0.93	198

Fig 10: Cross Validation



	precision	recall	f1-score	support
0	1.00	1.00	1.00	63
1	1.00	1.00	1.00	62
2	1.00	1.00	1.00	73
accuracy			1.00	198
macro avg	1.00	1.00	1.00	198
weighted avg	1.00	1.00	1.00	198

Fig 10: Decision Tree

4. Conclusion and Future Recommendations

The study started off by analyzing the data and discovering that some features, including age, appeared to be reliable predictors of cancer. Following considerable visualization, the study tried several Machine Learning approaches such as Cross Validation and Decision Tree. The models were then subjected to hyper parameter tuning to examine if the outcomes might be improved. The Decision Tree algorithm offered the best accuracy and F1 score. The study carefully examined the most logical methodologies and algorithms, stating their advancements, to comprehend the accuracy and recall prediction of the selected model about their F1 measurement. According to the study, the Decision Tree algorithm performed the best and was selected as the project's preferred model.

Appendixes

```
import numpy as np

import pandas as pd

import plotly.graph_objs as go

import plotly.figure_factory as ff

import matplotlib.pyplot as plt

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import confusion_matrix

from sklearn.tree import DecisionTreeClassifier

from sklearn import tree
```

```
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import GridSearchCV
from imblearn.over_sampling import SMOTE
from imblearn.pipeline import Pipeline
from sklearn.linear_model import SGDClassifier
from sklearn.metrics import classification_report
import warnings
warnings.filterwarnings("ignore", category=FutureWarning)
dataset = pd.read_excel("D:\AIUB\Spring_22-23\Final\ML\Final Project\cancer patient data sets.xlsx")
dataset.head()
print(dataset.shape)
dataset.describe().T
dataset.info()
dataset.drop(['Patient Id'], axis=1, inplace=True)
dataset.head()
def converter(column):
    if column == 'Low':
        return 0
    elif column == 'Medium':
        return 1
    else:
        return 2
dataset['Level']=dataset['Level'].apply(converter)
dataset.info()
dataset.isnull().sum()
columns=list(dataset.columns)
dataset[columns].plot(kind="box", subplots="True", layout=(5, 5) ,figsize=(10,8))
plt.show()
X=dataset.Age
Q1=np.quantile(X, .25)
Q3=np.quantile(X, .75)
IQR=Q3-Q1
Lb=Q1-(1.5*IQR)
Ub=Q3+(1.5*IQR)
Outliers=[]
for i in X:
```

```
    if (i<Lb or i> Ub):
        Outliers.append(i)
print("----------Outliers in variable 'Age' for the given dataset----------")
print(Outliers)
for ele in Outliers:
    dataset.drop(dataset[(dataset['Age']==ele)].index, inplace=True)
dataset.hist(figsize=(14,11))
plt.show()
corrs = dataset.corr()
figure = ff.create_annotated_heatmap(
    z=corrs.values,
    x=list(corrs.columns),
    y=list(corrs.index),
    annotation_text=corrs.round(2).values,
    showscale=True)
figure.show()
df_corr = dataset.corr()['Level'][:-1]
corr = df_corr[abs(df_corr) > 0.5].sort_values(ascending=False)
print("There is {} strongly correlated values with target:\n{}".format(len(corr), corr))
corr_df=pd.DataFrame(corr)
list(corr_df.index)
impColms=list(corr_df.index)
x=dataset.copy()
for i in dataset.
columns:
    if i not in impColms:
        x.drop([i],axis=1,inplace=True)
y=dataset['Level']
print(x.shape)
print(y.shape)
feature_scaler = StandardScaler()
X_scaled = feature_scaler.fit_transform(x)
X_scaled
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=132)
model = Pipeline([
```

```
        ('balancing', SMOTE(random_state = 107)),
        ('classification', SGDClassifier(loss = 'log', penalty = 'elasticnet', random_state = 38))
    ])
grid_param = {'classification__eta0': [.001,.01,.1,1,10,100], 'classification__max_iter' : [100,500,1000], 'classification__alpha': [.001, .01,.1, 1,10,100], 'classification__l1_ratio': [0,0.3,0.5,0.7,1]}
gd_sr = GridSearchCV(estimator=model, param_grid=grid_param, scoring='accuracy', cv=5)
gd_sr.fit(X_train,y_train)
best_parameters = gd_sr.best_params_
print("Best parameters: ", best_parameters)
best_result = gd_sr.best_score_
print("Best result: ", best_result)
mat_gs=confusion_matrix(y_test,y_pred_CV)
fig, ax = plt.subplots(figsize=(5, 5))
ax.matshow(mat_gs, cmap=plt.cm.Oranges, alpha=0.3)
for i in range(mat_gs.shape[0]):
    for j in range(mat_gs.shape[1]):
        ax.text(x=j, y=i,s=mat_gs[i, j], va='center', ha='center', size='xx-large')
plt.xlabel('Predictions', fontsize=18)
plt.ylabel('Actuals', fontsize=18)
plt.title('Confusion Matrix', fontsize=18)
plt.show()


print(classification_report(y_test, y_pred_CV))
DT = tree.DecisionTreeClassifier()
DT = DT.fit(X_train,y_train)
y_pred_DT = DT.predict(X_test)
print('Accuracy: {}'.format(DT.score(X_test, y_test)))
mat_dt=confusion_matrix(y_test,y_pred_DT)
fig, ax = plt.subplots(figsize=(5, 5))
ax.matshow(mat_dt, cmap=plt.cm.Oranges, alpha=0.3)
for i in range(mat_dt.shape[0]):
    for j in range(mat_dt.shape[1]):
        ax.text(x=j, y=i,s=mat_dt[i, j], va='center', ha='center', size='xx-large')
plt.xlabel('Predictions', fontsize=18)
plt.ylabel('Actuals', fontsize=18)
plt.title('Confusion Matrix', fontsize=18)
plt.show()
```

```
print(classification_report(y_test, y_pred_DT))
```

References:

1. https://www.kaggle.com/datasets/rishidamarla/cancer-patients-data/code
2. https://www.kaggle.com/code/raghadabdulhadi/lung-cancer
3. https://www.kaggle.com/datasets/thedevastator/cancer-patients-and-air-pollution-a-new-link
4. https://www.kaggle.com/code/adityahanwat/random-forest-and-logistic-regression-model