BCS2313
ARTIFICIAL INTELLIGENCE TECHNIQUES
(NEURAL NETWORK PROJECT)

SESSION 2023/2024 SEMESTER II

LECTURER'S NAME:TS DR. ANIS FARIHAN BINTI MAT RAFFEI

SECTION : 2

GROUP MEMBERS AND NO.MATRIC:

1. **MUHAMMAD ZAKI BIN RAHIM (CA21043)**
2. **NUR ILI LIYANA BINTI MOHD DAGAN (CD21017)**
3. **MUHAMAD ALIFF AIMAN BIN SHAHNI (CA21036)**

DATASET NAME : LUNG CANCER PREDICTION

Table of Contents

DATASET INFORMATION

This dataset contains information on patients with lung cancer, including their age, gender, air pollution exposure, 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, shortness of breath, wheezing, swallowing difficulty, clubbing of finger nails and snoring.

1.1 Objective

1. To develop and implement a robust neural network based system for the early detection and prediction of lung cancer.
2. To analyze medical imaging such as CT scans along with integrating clinical and histopathological data.
3. to create a comprehensive, user-friendly tool that can be seamlessly integrated into clinical workflows, providing clinicians with a powerful decision support system that enhances diagnostic precision and supports timely intervention.

1.2 Attributes

The attributes involved in this dataset include:

NO	INPUT	DESCRIPTION
1.	Age	The age of the patient.
2.	Gender	The gender of the patient.
3.	Air Pollution	The level of air pollution exposure of the patient.
4.	Alcohol Use	The level of alcohol use of the patient
5.	Dust Allergy	The level of dust allergy of the patient.
6.	OccuPational Hazards	The level of occupational hazards of the patient
7.	Genetic Risk	The level of genetic risk of the patient.
8.	Chronic Lung Disease	The level of chronic lung disease of the patient.
9.	Balanced Diet	The level of a balanced diet of the patient.
10	Obesity	The level of obesity of the patient.
11	Smoking	The level of smoking of the patient.
12	Passive Smoker	The level of passive smoker of the patient.
13	Chest Pain	The level of chest pain of the patient.
14	Coughing of Blood	The level of coughing of blood of the patient.
15	Fatigue	The level of fatigue of the patient.

16	Weight Loss	The level of weight loss of the patient.
17	Shortness of Breath	The level of shortness of breath of the patient.
18	Wheezing	The level of wheezing of the patient.
19	Swallowing Difficulty	The level of swallowing difficulty of the patient.
20	Clubbing of Finger Nails	The level of clubbing of finger nails of the patient.

NO	OUTPUT	DESCRIPTION
1	Level	The risk level for lung cancer ● High ● Medium ● Low

1.3 Total Dataset

The total number of datasets involved are 1001.

For this project, we had decided to take 304 data from the dataset for the purpose of developing a Neural Network System. The attributes involved in our system will be taken partially from the original attributes which are the age, gender, air pollution exposure, 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 ,shortness of breath ,wheezing ,swallowing difficulty ,clubbing of finger nails and snoring.

From our research and observation, the majority of lung cancer cases are attributed to smoking, but exposure to air pollution is also a risk factor. A new study has found that air pollution may be linked to an increased risk of lung cancer, even in nonsmokers.

1.4 Screenshot sample list of dataset

index	Patient Id	Age	Gender	Air Polluti	Alcohol us	Dust Aller	Occupatio	Genetic R	chronic Lu	Balanced I	Obesity	Smoking	Passive Sr	Chest Pair	Coughing	Fatigue	Weight Lo	Shortness	Wheezing	Swallowir	Clubbing (	Frequent	Dry Cough	Snoring	Level
0	P1	33	1	2	4	5	4	3	2	2	4	3	2	2	4	3	4	2	2	3	1	2	3	4	Low
1	P10	17	1	3	1	5	3	4	2	2	2	2	4	2	3	1	3	7	8	6	2	1	7	2	Medium
2	P100	35	1	4	5	6	5	5	4	6	7	2	3	4	8	8	7	9	2	1	4	6	7	2	High
3	P1000	37	1	7	7	7	7	6	7	7	7	7	7	7	8	4	2	3	1	4	5	6	7	5	High
4	P101	46	1	6	8	7	7	7	6	7	7	8	7	7	9	3	2	4	1	4	2	4	2	3	High
5	P102	35	1	4	5	6	5	5	4	6	7	2	3	4	8	8	7	9	2	1	4	6	7	2	High
6	P103	52	2	2	4	5	4	3	2	2	4	3	2	2	4	3	4	2	2	3	1	2	3	4	Low
7	P104	28	2	3	1	4	3	2	3	4	3	1	4	3	1	3	2	2	4	2	2	3	4	3	Low
8	P105	35	2	4	5	6	5	6	5	5	5	6	6	6	5	1	4	3	2	4	6	2	4	1	Medium
9	P106	46	1	2	3	4	2	4	3	3	3	2	3	4	4	1	2	4	6	5	4	2	1	5	Medium
10	P107	44	1	6	7	7	7	7	6	7	7	7	8	7	7	5	3	2	7	8	2	4	5	3	High
11	P108	64	2	6	8	7	7	7	6	7	7	7	8	7	7	9	6	5	7	2	4	3	1	4	High
12	P109	39	2	4	5	6	6	5	4	6	6	6	6	6	6	5	3	2	4	3	1	7	5	6	Medium
13	P11	34	1	6	7	7	7	6	7	7	7	7	7	7	8	4	2	3	1	4	5	6	7	5	High
14	P110	27	2	3	1	4	2	3	2	3	3	2	2	4	2	2	2	3	4	1	5	2	6	2	Low
15	P111	73	1	5	6	6	5	6	5	6	5	8	5	5	5	4	3	6	2	1	2	1	6	2	Medium
16	P112	17	1	3	1	5	3	4	2	2	2	2	4	2	3	1	3	7	8	6	2	1	7	2	Medium
17	P113	34	1	6	7	7	7	6	7	7	7	7	7	7	8	4	2	3	1	4	5	6	7	5	High
18	P114	36	1	6	7	7	7	7	6	7	7	7	7	7	8	8	5	7	6	7	8	7	6	2	High
19	P115	14	1	2	4	5	6	5	5	4	6	5	4	6	5	5	3	2	1	4	7	2	1	6	Medium
20	P116	24	1	6	8	7	7	6	7	7	3	8	7	9	6	5	2	5	2	3	2	1	7	6	High
21	P117	53	2	4	5	6	5	5	4	6	7	2	3	4	8	8	7	9	2	1	4	6	7	2	High
22	P118	62	1	6	8	7	7	7	6	7	7	8	7	7	9	3	2	4	1	4	2	4	2	3	High
23	P119	29	2	6	7	7	7	7	6	7	7	7	7	7	7	2	7	6	7	6	7	2	3	1	High
24	P12	36	1	6	7	7	7	7	6	7	7	7	7	7	7	8	5	7	6	7	8	7	6	2	High
25	P120	65	1	6	8	7	7	7	6	2	4	1	2	4	3	2	7	6	5	1	9	3	4	2	Medium
26	P121	38	2	2	1	5	3	2	3	2	4	1	4	2	4	6	7	2	5	8	1	3	2	3	Medium
27	P122	19	1	3	2	4	2	3	2	3	3	2	2	3	3	4	5	6	5	5	4	6	5	4	Medium
28	P123	33	1	6	7	7	7	7	6	7	7	4	8	7	7	4	4	5	6	5	5	4	6	5	High
29	P124	28	2	1	6	7	5	3	2	6	2	3	3	2	2	3	3	7	7	4	8	7	7	5	Medium
30	P125	35	2	2	6	2	3	6	6	6	4	6	8	7	6	5	5	4	6	5	4	6	5	7	High
31	P126	42	1	2	4	5	6	5	5	4	6	7	7	2	3	8	7	7	3	8	9	1	6	2	High
32	P127	32	2	1	6	7	8	7	6	7	7	3	4	8	7	3	2	6	4	2	3	1	2	1	Medium
33	P128	33	1	2	4	5	4	3	2	2	4	3	2	2	4	3	4	2	2	3	1	2	3	4	Low
34	P129	25	2	3	1	4	3	2	3	4	3	1	4	3	1	3	2	2	4	2	2	3	4	3	Low
35	P13	14	1	2	4	5	6	5	5	4	6	5	4	6	5	5	3	2	1	4	7	2	1	6	Medium
36	P130	27	2	3	1	4	2	3	2	3	3	2	2	4	2	2	2	3	4	1	5	2	6	2	Low
37	P131	28	1	6	7	8	7	6	7	7	2	4	3	7	8	2	3	6	4	2	3	1	2	1	Low
38	P132	32	1	2	3	6	7	7	7	7	2	4	3	7	4	2	1	3	2	2	1	2	5	1	Low
39	P133	45	2	1	2	4	5	6	5	5	4	6	4	7	2	3	8	7	3	8	3	2	3	1	Low
40	P134	27	2	3	1	4	3	2	3	4	3	1	4	3	1	3	2	2	4	2	2	3	4	3	Low
41	P135	26	2	2	3	1	4	3	2	3	4	3	1	4	3	1	2	3	4	5	1	2	3	4	Low
42	P136	48	1	4	2	3	2	1	2	3	2	1	5	1	4	2	6	1	2	4	2	1	2	3	Low
43	P137	17	2	1	2	3	4	4	3	2	1	3	2	1	2	1	3	3	2	1	3	2	1	1	Low
44	P138	22	1	2	1	3	4	3	5	3	2	6	1	1	2	3	2	1	3	2	4	2	1	1	Low
45	P139	42	1	2	1	2	3	4	3	2	1	1	1	6	2	1	1	1	2	1	2	3	1	2	Low
46	P14	24	1	6	8	7	7	6	7	7	3	8	7	9	6	5	2	5	2	3	2	1	7	6	High
47	P140	35	1	1	3	2	4	2	6	2	2	2	1	3	4	4	2	2	2	3	2	1	2	4	Low
48	P141	24	2	1	2	2	3	2	4	2	3	2	1	1	1	1	1	2	3	4	5	2	1	2	Low
49	P142	38	2	3	2	3	2	3	2	3	2	3	2	1	1	1	2	3	2	5	1	5	1	1	Low
50	P143	18	2	3	2	1	3	2	1	3	2	1	2	2	2	2	2	1	3	4	4	1	4	1	Low
51	P144	23	2	4	2	3	4	2	3	2	4	2	4	2	4	1	3	4	2	4	2	4	3	1	Low
52	P145	24	2	3	2	2	1	1	1	1	1	4	2	3	6	2	1	2	3	4	2	1	1	1	Low
53	P146	35	2	2	1	2	1	2	1	2	3	2	4	2	1	3	4	5	1	3	2	1	2	2	Low
54	P147	38	2	5	2	3	1	2	3	5	2	2	5	1	3	1	1	1	1	1	3	2	4	2	Low
55	P148	47	2	2	3	1	3	2	5	2	1	2	1	2	5	3	2	1	2	3	1	3	4	2	Low
56	P149	52	2	3	2	1	2	3	5	1	2	7	2	1	1	1	1	1	3	2	3	2	3	3	Low
57	P15	53	2	4	5	6	5	5	4	6	7	2	3	4	8	8	7	9	2	1	4	6	7	2	High
58	P150	44	2	2	3	2	1	3	2	1	2	7	6	2	2	2	2	3	2	1	2	3	2	3	Low
59	P151	38	2	2	3	5	2	1	1	1	1	4	3	2	4	2	1	3	4	6	1	3	2	2	Low
60	P152	62	2	3	2	1	3	2	4	5	1	6	2	3	2	4	3	2	1	2	4	2	3	2	Low
61	P153	61	2	2	3	4	2	1	1	2	4	3	2	1	5	2	1	3	2	1	3	2	1	2	Low
62	P154	55	1	3	1	1	1	2	3	4	1	3	2	4	3	2	5	2	1	2	3	4	5	2	Low
63	P155	45	2	1	2	3	4	2	4	3	3	3	2	3	4	4	1	2	4	6	5	4	2	5	Medium
64	P156	38	1	6	8	7	7	7	6	2	4	1	2	4	3	2	7	6	5	1	9	3	4	2	Medium
65	P157	44	1	2	1	5	3	2	3	2	4	1	4	2	4	6	7	2	5	8	1	3	2	3	Medium
66	P158	45	2	3	2	4	2	3	2	3	3	2	2	3	3	4	5	6	5	5	4	6	5	4	Medium
67	P159	33	2	1	6	7	8	7	6	7	7	3	4	8	7	3	2	6	4	2	3	1	2	1	Medium
68	P16	62	1	6	8	7	7	7	6	7	7	8	7	7	9	3	2	4	1	4	2	4	2	3	High
69	P160	32	2	1	6	7	5	3	2	6	2	3	3	2	2	3	3	7	7	4	8	7	7	5	Medium
70	P161	44	1	1	2	3	4	2	4	3	3	3	2	3	4	4	1	2	4	6	5	4	2	5	Medium
71	P162	62	1	6	8	7	7	7	6	2	4	1	2	4	3	2	7	6	5	1	9	3	4	2	Medium
72	P163	38	2	2	1	5	3	2	3	2	4	1	4	2	4	6	7	2	5	8	1	3	2	3	Medium

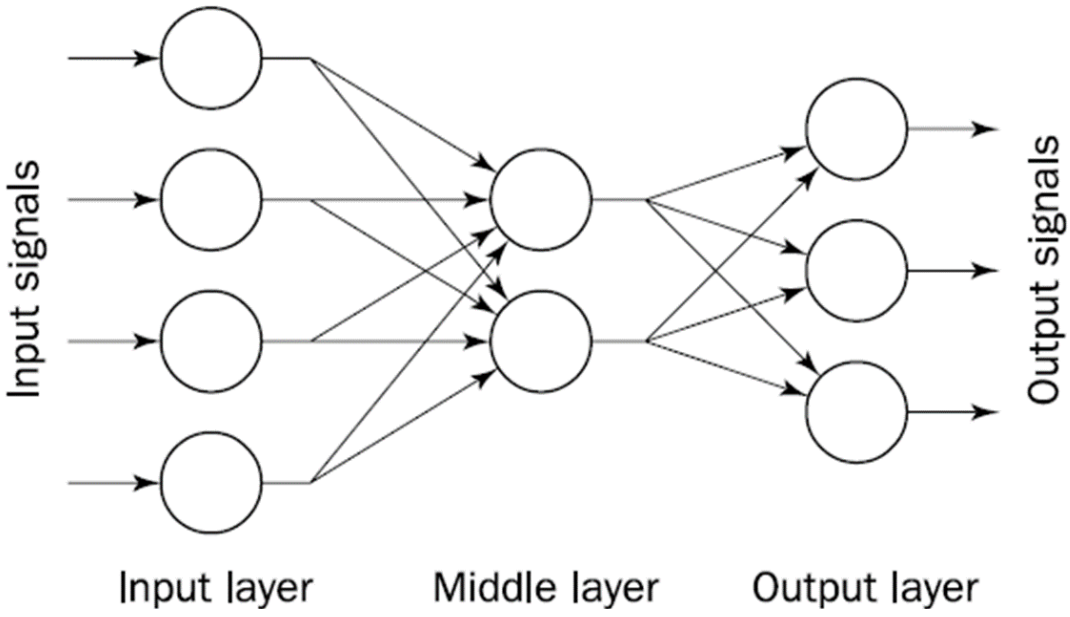
75	73	P164	33	1	3	2	4	2	3	2	3	3	2	2	3	3	4	5	6	5	5	4	6	5	4	Medium
76	74	P165	22	1	6	8	7	7	7	6	2	4	1	2	4	3	2	7	6	5	1	9	3	4	2	Medium
77	75	P166	35	1	2	1	5	3	2	3	2	4	1	4	2	4	6	7	2	5	8	1	3	2	3	Medium
78	76	P167	23	1	3	2	4	2	3	2	3	3	2	2	3	3	4	5	6	5	5	4	6	5	4	Medium
79	77	P168	48	1	1	6	7	8	7	6	7	7	3	4	8	7	3	2	6	4	2	3	1	2	1	Medium
80	78	P169	46	2	1	6	7	5	3	2	6	2	3	3	2	2	3	3	7	7	4	8	7	7	5	Medium
81	79	P17	29	2	6	7	7	7	7	6	7	7	7	7	7	7	2	7	6	7	6	7	2	3	1	High
82	80	P170	52	2	1	2	3	4	2	4	3	3	3	2	3	4	4	1	2	4	6	5	4	2	5	Medium
83	81	P171	52	1	6	8	7	7	7	6	2	4	1	2	4	3	2	7	6	5	1	9	3	4	2	Medium
84	82	P172	48	2	6	8	7	7	7	6	2	4	1	2	4	3	2	7	6	5	1	9	3	4	2	Medium
85	83	P173	36	2	2	1	5	3	2	3	2	4	1	4	2	4	6	7	2	5	8	1	3	2	3	Medium
86	84	P174	31	2	3	2	4	2	3	2	3	3	2	2	3	3	4	5	6	5	5	4	6	5	4	Medium
87	85	P175	38	2	1	2	3	4	2	4	3	3	3	2	3	4	4	1	2	4	6	5	4	2	5	Medium
88	86	P176	35	1	6	8	7	7	7	6	2	4	1	2	4	3	2	7	6	5	1	9	3	4	2	Medium
89	87	P177	44	1	6	7	7	7	7	6	7	7	7	8	7	7	5	3	2	7	8	2	4	5	3	High
90	88	P178	33	1	6	8	7	7	7	6	7	7	7	8	7	7	9	6	5	7	2	4	3	1	4	High
91	89	P179	45	1	6	7	7	7	7	6	7	7	4	8	7	7	4	4	5	6	5	5	4	6	5	High
92	90	P18	65	1	6	8	7	7	7	6	2	4	1	2	4	3	2	7	6	5	1	9	3	4	2	Medium
93	91	P180	53	1	6	8	7	7	6	7	7	3	8	7	9	6	5	2	5	2	3	2	1	7	6	High
94	92	P181	35	2	4	5	6	5	5	4	6	7	2	3	4	8	8	7	9	2	1	4	6	7	2	High
95	93	P182	46	2	6	8	7	7	7	6	7	7	8	7	7	9	3	2	4	1	4	2	4	2	3	High
96	94	P183	27	1	6	7	7	7	7	6	7	7	7	7	7	7	2	7	6	7	6	7	2	3	1	High
97	95	P184	26	1	6	7	7	7	6	7	7	7	7	7	7	8	4	2	3	1	4	5	6	7	5	High
98	96	P185	37	1	6	7	7	7	7	6	7	7	7	7	7	7	8	5	7	6	7	8	7	6	2	High
99	97	P186	28	1	6	7	7	7	7	6	7	7	7	8	7	7	5	3	2	7	8	2	4	5	3	High
100	98	P187	19	1	6	8	7	7	7	6	7	7	7	8	7	7	9	6	5	7	2	4	3	1	4	High
101	99	P188	29	2	4	5	6	5	5	4	6	7	2	3	4	8	8	7	9	2	1	4	6	7	2	High
102	100	P189	39	2	6	8	7	7	7	6	7	7	8	7	7	9	3	2	4	1	4	2	4	2	3	High
103	101	P19	38	2	2	1	5	3	2	3	2	4	1	4	2	4	6	7	2	5	8	1	3	2	3	Medium
104	102	P190	49	1	6	5	6	5	5	4	6	7	2	3	4	8	8	7	9	2	1	4	6	7	2	High
105	103	P191	37	1	8	8	7	7	7	6	7	7	8	7	7	9	3	2	4	1	4	2	4	2	3	High
106	104	P192	26	2	7	7	7	7	7	6	7	7	7	7	7	7	2	7	6	7	6	7	2	3	1	High
107	105	P193	37	2	7	7	7	7	6	7	7	7	7	7	7	8	4	2	3	1	4	5	6	7	5	High
108	106	P194	33	1	6	7	7	7	7	7	6	7	7	7	7	7	8	5	7	6	7	8	7	6	2	High
109	107	P195	44	1	6	7	7	7	7	6	7	7	7	8	7	7	5	3	2	7	8	2	4	5	3	High
110	108	P196	37	2	6	8	7	7	7	6	7	7	7	8	7	7	9	6	5	7	2	4	3	1	4	High
111	109	P197	25	2	4	5	6	5	5	4	6	7	2	3	4	8	8	7	9	2	1	4	6	7	2	High
112	110	P198	18	2	6	8	7	7	7	6	7	7	8	7	7	9	3	2	4	1	4	2	4	2	3	High

111	P199	47	1	6	5	6	5	5	4	6	7	2	3	4	8	8	7	9	2	1	4	6	7	2	High
112	P2	25	2	3	1	4	3	2	3	4	3	1	4	3	1	3	2	2	4	2	2	3	4	3	Low
113	P20	19	1	3	2	4	2	3	2	3	3	2	2	3	3	4	5	6	5	5	4	6	5	4	Medium
114	P200	26	2	8	8	7	7	7	6	7	7	8	7	7	9	3	2	4	1	4	2	4	2	3	High
115	P201	37	1	7	7	7	7	6	7	7	7	7	7	7	8	4	2	3	1	4	5	6	7	5	High
116	P202	35	2	4	5	6	5	5	4	6	7	2	3	4	8	8	7	9	2	1	4	6	7	2	High
117	P203	33	1	2	4	5	4	3	2	2	4	3	2	2	4	3	4	2	2	3	1	2	3	4	Low
118	P204	25	2	3	1	4	3	2	3	4	3	1	4	3	1	3	2	2	4	2	2	3	4	3	Low
119	P205	35	2	4	5	6	5	6	5	5	5	6	6	6	5	1	4	3	2	4	6	2	4	1	Medium
120	P206	27	2	2	3	4	2	4	3	3	3	2	3	4	4	1	2	4	6	5	4	2	1	5	Medium
121	P207	48	1	6	7	7	7	7	6	7	7	7	8	7	7	5	3	2	7	8	2	4	5	3	High
122	P208	64	1	6	8	7	7	7	6	7	7	7	8	7	7	9	6	5	7	2	4	3	1	4	High
123	P209	39	1	4	5	6	6	5	4	6	6	6	6	6	6	5	3	2	4	3	1	7	5	6	Medium
124	P21	33	1	6	7	7	7	7	6	7	7	4	8	7	7	4	4	5	6	5	5	4	6	5	High
125	P210	27	2	3	1	4	2	3	2	3	3	2	2	4	2	2	2	3	4	1	5	2	6	2	Low
126	P211	73	1	5	6	6	5	6	5	6	5	8	5	5	5	4	3	6	2	1	2	1	6	2	Medium
127	P212	17	1	3	1	5	3	4	2	2	2	2	4	2	3	1	3	7	8	6	2	1	7	2	Medium
128	P213	34	1	6	7	7	7	6	7	7	7	7	7	7	8	4	2	3	1	4	5	6	7	5	High
129	P214	36	1	6	7	7	7	7	7	6	7	7	7	7	7	8	5	7	6	7	8	7	6	2	High
130	P215	14	1	2	4	5	6	5	5	4	6	5	4	6	5	5	3	2	1	4	7	2	1	6	Medium
131	P216	24	1	6	8	7	7	6	7	7	3	8	7	9	6	5	2	5	2	3	2	1	7	6	High
132	P217	53	2	4	5	6	5	5	4	6	7	2	3	4	8	8	7	9	2	1	4	6	7	2	High
133	P218	62	1	6	8	7	7	7	6	7	7	8	7	7	9	3	2	4	1	4	2	4	2	3	High
134	P219	29	2	6	7	7	7	7	6	7	7	7	7	7	7	2	7	6	7	6	7	2	3	1	High
135	P22	28	2	1	6	7	5	3	2	6	2	3	3	2	2	3	3	7	7	4	8	7	7	5	Medium
136	P220	65	1	6	8	7	7	7	6	2	4	1	2	4	3	2	7	6	5	1	9	3	4	2	Medium
137	P221	38	2	2	1	5	3	2	3	2	4	1	4	2	4	6	7	2	5	8	1	3	2	3	Medium
138	P222	19	1	3	2	4	2	3	2	3	3	2	2	3	3	4	5	6	5	5	4	6	5	4	Medium
139	P223	33	1	6	7	7	7	7	6	7	7	4	8	7	7	4	4	5	6	5	5	4	6	5	High
140	P224	28	2	1	6	7	5	3	2	6	2	3	3	2	2	3	3	7	7	4	8	7	7	5	Medium
141	P225	35	2	2	6	2	3	6	6	6	4	6	8	7	6	5	5	4	6	5	4	6	5	7	High
142	P226	42	1	2	4	5	6	5	5	4	6	7	7	2	3	8	7	7	3	8	9	1	6	2	High
143	P227	32	2	1	6	7	8	7	6	7	7	3	4	8	7	3	2	6	4	2	3	1	2	1	Medium
144	P228	33	1	2	4	5	4	3	2	2	4	3	2	2	4	3	4	2	2	3	1	2	3	4	Low
145	P229	25	2	3	1	4	3	2	3	4	3	1	4	3	1	3	2	2	4	2	2	3	4	3	Low
146	P23	35	2	2	6	2	3	6	6	6	4	6	8	7	6	5	5	4	6	5	4	6	5	7	High
147	P230	27	2	3	1	4	2	3	2	3	3	2	2	4	2	2	2	3	4	1	5	2	6	2	Low
148	P231	28	1	6	7	8	7	6	7	7	2	4	3	7	8	2	3	6	4	2	3	1	2	1	Low

149	P232	32	1	2	3	6	7	7	7	7	2	4	3	7	4	2	1	3	2	2	1	2	5	1	Low
150	P233	45	2	1	2	4	5	6	5	5	4	6	4	7	2	3	8	7	3	8	3	2	3	1	Low
151	P234	27	1	3	1	4	3	2	3	4	3	1	4	3	1	3	2	2	4	2	2	3	4	3	Low
152	P235	26	1	2	3	1	4	3	2	3	4	3	1	4	3	1	2	3	4	5	1	2	3	4	Low
153	P236	48	1	4	2	3	2	1	2	3	2	1	5	1	4	2	6	1	2	4	2	1	2	3	Low
154	P237	17	2	1	2	3	4	4	3	2	1	3	2	1	2	1	3	3	2	1	3	2	1	1	Low
155	P238	22	2	2	1	3	4	3	5	3	2	6	1	1	2	3	2	1	3	2	4	2	1	1	Low
156	P239	42	1	2	1	2	3	4	3	2	1	1	1	6	2	1	1	1	2	1	2	3	1	2	Low
157	P24	42	1	2	4	5	6	5	5	4	6	7	7	2	3	8	7	7	3	8	9	1	6	2	High
158	P240	35	1	1	3	2	4	2	6	2	2	2	1	3	4	4	2	2	2	3	2	1	2	4	Low
159	P241	24	2	1	2	2	3	2	4	2	3	2	1	1	1	1	1	1	2	3	4	5	2	1	Low
160	P242	38	2	3	2	3	2	3	2	3	2	3	2	1	1	1	2	3	2	5	1	5	1	1	Low
161	P243	18	2	3	2	1	3	2	1	3	2	1	2	2	2	2	2	1	3	4	4	1	4	1	Low
162	P244	23	2	4	2	3	4	2	3	2	4	2	4	2	4	1	3	4	2	4	2	4	3	1	Low
163	P245	24	2	3	2	2	1	1	1	1	1	4	2	3	6	2	1	2	3	4	2	1	1	1	Low
164	P246	35	2	2	1	2	1	2	1	2	3	2	4	2	1	3	4	5	1	3	2	1	2	2	Low
165	P247	38	2	5	2	3	1	2	3	5	2	2	5	1	3	1	1	1	1	1	3	2	4	2	Low
166	P248	47	2	2	3	1	3	2	5	2	1	2	1	2	5	3	2	1	2	3	1	3	4	2	Low
167	P249	52	1	3	2	1	2	3	5	1	2	7	2	1	1	1	1	1	3	2	3	2	3	3	Low
168	P25	32	2	1	6	7	8	7	6	7	7	3	4	8	7	3	2	6	4	2	3	1	2	1	Medium
169	P250	44	1	2	3	2	1	3	2	1	2	7	6	2	2	2	2	3	2	1	2	3	2	3	Low
170	P251	38	1	2	3	5	2	1	1	1	1	4	3	2	4	2	1	3	4	6	1	3	2	2	Low
171	P252	62	1	3	2	1	3	2	4	5	1	6	2	3	2	4	3	2	1	2	4	2	3	2	Low
172	P253	61	1	2	3	4	2	1	1	2	4	3	2	1	5	2	1	3	2	1	3	2	1	2	Low
173	P254	55	1	3	1	1	1	2	3	4	1	3	2	4	3	2	5	2	1	2	3	4	5	2	Low
174	P255	45	2	1	2	3	4	2	4	3	3	3	2	3	4	4	1	2	4	6	5	4	2	5	Medium
175	P256	38	2	6	8	7	7	7	6	2	4	1	2	4	3	2	7	6	5	1	9	3	4	2	Medium
176	P257	44	1	2	1	5	3	2	3	2	4	1	4	2	4	6	7	2	5	8	1	3	2	3	Medium
177	P258	45	1	3	2	4	2	3	2	3	3	2	2	3	3	4	5	6	5	5	4	6	5	4	Medium
178	P259	33	1	1	6	7	8	7	6	7	7	3	4	8	7	3	2	6	4	2	3	1	2	1	Medium
179	P26	33	1	2	4	5	4	3	2	2	4	3	2	2	4	3	4	2	2	3	1	2	3	4	Low
180	P260	32	1	1	6	7	5	3	2	6	2	3	3	2	2	3	3	7	7	4	8	7	7	5	Medium
181	P261	44	1	1	2	3	4	2	4	3	3	3	2	3	4	4	1	2	4	6	5	4	2	5	Medium
182	P262	62	1	6	8	7	7	7	6	2	4	1	2	4	3	2	7	6	5	1	9	3	4	2	Medium
183	P263	38	2	2	1	5	3	2	3	2	4	1	4	2	4	6	7	2	5	8	1	3	2	3	Medium
184	P264	33	1	3	2	4	2	3	2	3	3	2	2	3	3	4	5	6	5	5	4	6	5	4	Medium
185	P265	22	1	6	8	7	7	7	6	2	4	1	2	4	3	2	7	6	5	1	9	3	4	2	Medium
186	P266	35	1	2	1	5	3	2	3	2	4	1	4	2	4	6	7	2	5	8	1	3	2	3	Medium
187	P267	23	1	3	2	4	2	3	2	3	3	2	2	3	3	4	5	6	5	5	4	6	5	4	Medium
188	P268	48	2	1	6	7	8	7	6	7	7	3	4	8	7	3	2	6	4	2	3	1	2	1	Medium
189	P269	46	2	1	6	7	5	3	2	6	2	3	3	2	2	3	3	7	7	4	8	7	7	5	Medium
190	P27	25	2	3	1	4	3	2	3	4	3	1	4	3	1	3	2	2	4	2	2	3	4	3	Low
191	P270	52	2	1	2	3	4	2	4	3	3	3	2	3	4	4	1	2	4	6	5	4	2	5	Medium
192	P271	52	1	6	8	7	7	7	6	2	4	1	2	4	3	2	7	6	5	1	9	3	4	2	Medium
193	P272	48	2	6	8	7	7	7	6	2	4	1	2	4	3	2	7	6	5	1	9	3	4	2	Medium
194	P273	36	2	2	1	5	3	2	3	2	4	1	4	2	4	6	7	2	5	8	1	3	2	3	Medium
195	P274	31	2	3	2	4	2	3	2	3	3	2	2	3	3	4	5	6	5	5	4	6	5	4	Medium
196	P275	38	2	1	2	3	4	2	4	3	3	3	2	3	4	4	1	2	4	6	5	4	2	5	Medium
197	P276	35	1	6	8	7	7	7	6	2	4	1	2	4	3	2	7	6	5	1	9	3	4	2	Medium
198	P277	44	1	6	7	7	7	7	6	7	7	7	8	7	7	5	3	2	7	8	2	4	5	3	High
199	P278	33	1	6	8	7	7	7	6	7	7	7	8	7	7	9	6	5	7	2	4	3	1	4	High
200	P279	45	1	6	7	7	7	7	6	7	7	4	8	7	7	4	4	5	6	5	5	4	6	5	High
201	P28	27	2	3	1	4	2	3	2	3	3	2	2	4	2	2	2	3	4	1	5	2	6	2	Low
202	P280	53	1	6	8	7	7	6	7	7	3	8	7	9	6	5	2	5	2	3	2	1	7	6	High
203	P281	35	2	4	5	6	5	5	4	6	7	2	3	4	8	8	7	9	2	1	4	6	7	2	High
204	P282	46	1	6	8	7	7	7	6	7	7	8	7	7	9	3	2	4	1	4	2	4	2	3	High
205	P283	27	1	6	7	7	7	7	6	7	7	7	7	7	7	2	7	6	7	6	7	2	3	1	High
206	P284	26	1	6	7	7	7	6	7	7	7	7	7	7	8	4	2	3	1	4	5	6	7	5	High
207	P285	37	1	6	7	7	7	7	7	6	7	7	7	7	7	8	5	7	6	7	8	7	6	2	High
208	P286	28	1	6	7	7	7	7	6	7	7	7	8	7	7	5	3	2	7	8	2	4	5	3	High
209	P287	19	1	6	8	7	7	7	6	7	7	7	8	7	7	9	6	5	7	2	4	3	1	4	High
210	P288	29	2	4	5	6	5	5	4	6	7	2	3	4	8	8	7	9	2	1	4	6	7	2	High
211	P289	39	2	6	8	7	7	7	6	7	7	8	7	7	9	3	2	4	1	4	2	4	2	3	High
212	P29	28	1	6	7	8	7	6	7	7	2	4	3	7	8	2	3	6	4	2	3	1	2	1	Low
213	P290	49	1	6	5	6	5	5	4	6	7	2	3	4	8	8	7	9	2	1	4	6	7	2	High
214	P291	37	1	8	8	7	7	7	6	7	7	8	7	7	9	3	2	4	1	4	2	4	2	3	High
215	P292	26	2	7	7	7	7	7	6	7	7	7	7	7	7	2	7	6	7	6	7	2	3	1	High
216	P293	37	2	7	7	7	7	6	7	7	7	7	7	7	8	4	2	3	1	4	5	6	7	5	High
217	P294	33	1	6	7	7	7	7	7	6	7	7	7	7	7	8	5	7	6	7	8	7	6	2	High
218	P295	44	1	6	7	7	7	7	6	7	7	7	8	7	7	5	3	2	7	8	2	4	5	3	High
219	P296	37	2	6	8	7	7	7	6	7	7	7	8	7	7	9	6	5	7	2	4	3	1	4	High
220	P297	25	2	4	5	6	5	5	4	6	7	2	3	4	8	8	7	9	2	1	4	6	7	2	High
221	P298	18	2	6	8	7	7	7	6	7	7	8	7	7	9	3	2	4	1	4	2	4	2	3	High
222	P299	47	1	6	5	6	5	5	4	6	7	2	3	4	8	8	7	9	2	1	4	6	7	2	High
223	P3	35	1	4	5	6	5	6	5	5	5	6	6	6	5	1	4	3	2	4	6	2	4	1	Medium
224	P30	32	1	2	3	6	7	7	7	7	2	4	3	7	4	2	1	3	2	2	1	2	5	1	Low

METHODOLOGY

2.1 Neural network framework



Input Layer:

- The input layer represents the neural network's initial process, where it collects raw data for the process.
- Each neuron(Nodes) in the input layer represents one of the input data's features or dimensions.
- The input layer receives input signals and forwards them to the next layer, which is usually a hidden layer, without performing any calculation other than maybe scaling or normalizing the input data.

Middle Layer:

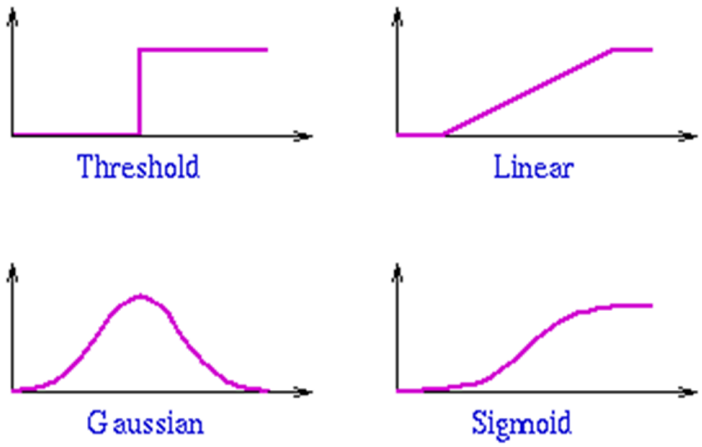
- The hidden layer is where the processing and learning occur. They are referred to be "hidden" because they are not directly visible in the input or output; they only function as mediators.
- Each neuron in the Middle/hidden layer model receives inputs from layers above and sends outputs to the layer below. A weight, a learnable parameter that determines the

direction and strength of effect, is associated with each neuronal connection. In addition, every neuron has a bias term, a learnable parameter that may be used to improve the data fit of the model by shifting the activation function to the left or right.

- Each neuron calculates a weight sum of its inputs as below:

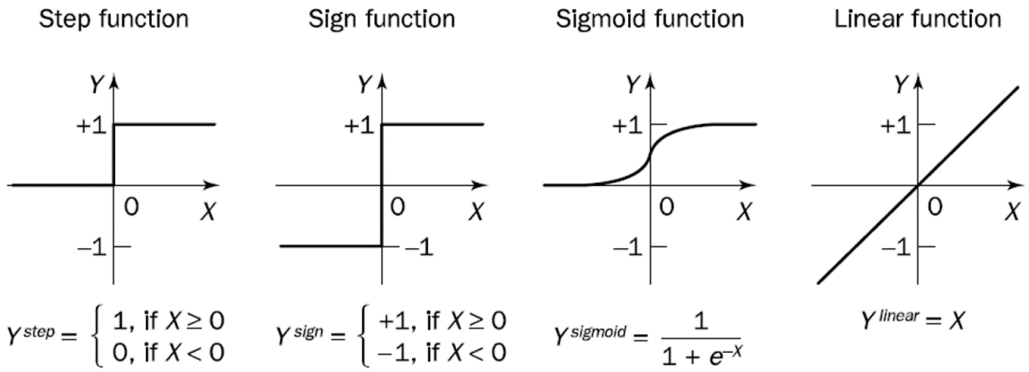
$$z_j = \sum_i (W_{ij} \cdot x_i) + b_j$$

- The activation function used to the weighted sum promotes non-linearity in the network, which is common and includes as below:



Output Layer

- The finalized outputs of a neural network, which are the classifications or predictions made from the input data, are produced by the output layer.
- A neural network's output layer calculates the weighted sum of its inputs, which is then transmitted through an activation function that is selected according to the task.



- The network's final output from the output layer is its prediction for classification tasks, the class with the highest probability is often selected.

2.2 Hyperparameters

This section explains the hyperparameters utilized in our neural network framework, specifically focusing on their functions and the criteria we employed to choose them. We focused on optimizing the primary hyperparameters, which include the learning rate, the number of epochs, and the batch size. The impact of each hyperparameter on the model's performance was assessed by evaluating them using two sets of values.

Learning Rate

Learning Rate Set 1: 0.001
Learning Rate Set 2: 0.01

The learning rate determines the magnitude of the adjustments made to the model's weights based on the loss gradient during each update. A lower learning rate, such as 0.001, results in a slower and more gradual learning process. This can be beneficial as it allows for the identification of a more precise minimum of the loss function. This pace is often consistent and prevents exceeding the best solution. Conversely, a higher learning rate, such as 0.01, speeds up the training process but carries the risk of the model converging too rapidly to a suboptimal point or even diverging if the step size is excessively big.

Epochs

Epochs Set 1: 50
Epochs Set 2: 100

An epoch refers to a single iteration over the full training dataset. Increasing the number of epochs enhances the model's ability to extract knowledge from the data. In the initial set, we employed 50 epochs, a moderate quantity that enables the model to acquire adequate knowledge without succumbing to overfitting. In the second batch, we augmented the number of epochs to

100, allowing the model additional opportunity to adapt its parameters and potentially attain superior performance. Nevertheless, a greater number of epochs also heightens the potential for overfitting if not adequately managed.

Batch Size

Batch Size Set 1: 32
Batch Size Set 2: 64

Definition: The batch size determines the quantity of samples that will be processed simultaneously by the network. Reducing the batch size to 32 causes the model to update its weights more often, resulting in a more stable learning process and improved generalization. Nevertheless, it can require a significant amount of processing resources. Increasing the batch size, such as to 64, enables faster training by fully utilizing the computational hardware. However, it might occasionally result in less stable convergence and inferior generalization to fresh data.

Hyperparameter	Set 1 Value	Set 2 Value	Explainantion
Learning Rate	0.001	0.01	Controls step size during gradient descent.
Epochs	50	100	Number of complete passes through the training data.
Batch Size	32	64	Number of samples processed before the model updates.

2.3 Source Code

```
import numpy as np
import pandas as pd
import tensorflow as tf
from matplotlib import pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score,
confusion_matrix, classification_report, mean_squared_error

# Load the dataset
file_path = 'cancer.csv'  # Ensure the correct file path
data = pd.read_csv(file_path)

# Display unique values in the original 'Level' column before encoding
unique_levels = data['Level'].unique()
print("Unique levels before encoding:", unique_levels)

# Encode categorical variables
label_encoder = LabelEncoder()
data['Gender'] = label_encoder.fit_transform(data['Gender'])

# Display the mapping of encoded values for 'Level'
encoded_levels = label_encoder.fit_transform(unique_levels)
for original, encoded in zip(unique_levels, encoded_levels):
    print(f"Original: {original}, Encoded: {encoded}")

data['Level'] = label_encoder.fit_transform(data['Level'])

# Features and target variable
```

```
X = data.drop(columns=['index', 'Patient Id', 'Level'])
y = data['Level']

# Plot class distribution
plt.figure(figsize=(8, 6))
sns.countplot(x=y)
plt.title('Class Distribution')
plt.xlabel('Class')
plt.ylabel('Frequency')
plt.show()

# Split the dataset 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)

# Normalize the features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Define the input layers for the model
inputs = {col: tf.keras.layers.Input(shape=(1,), dtype=tf.float32, name=col) for col in
X.columns}

# Concatenate the inputs into a single tensor
concatenated_inputs = tf.keras.layers.Concatenate()(list(inputs.values()))

# Define the plotting function
def plot_the_loss_curve(epochs, loss_training, loss_validation):
    plt.figure()
    plt.xlabel("Epoch")
    plt.ylabel("Loss")
```

```
    plt.plot(epochs, loss_training, label="Training Loss")
    plt.plot(epochs, loss_validation, label="Validation Loss")
    merged_loss_lists = loss_training.tolist() + loss_validation
    highest_loss = max(merged_loss_lists)
    lowest_loss = min(merged_loss_lists)
    top_of_y_axis = highest_loss * 1.03
    bottom_of_y_axis = lowest_loss * 0.97
    plt.ylim([bottom_of_y_axis, top_of_y_axis])
    plt.legend()
    plt.show()

# Define functions to create and train a model
def create_model(inputs, learning_rate, num_classes):
    x = tf.keras.layers.Dense(64, activation='relu')(inputs)  # Hidden layer with 64 units and ReLU
activation
    x = tf.keras.layers.Dense(32, activation='relu')(x)       # Hidden layer with 32 units and ReLU
activation
    outputs = tf.keras.layers.Dense(num_classes, activation='softmax')(x)  # Output layer with
softmax activation
    model = tf.keras.Model(inputs=inputs, outputs=outputs)
    model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=learning_rate),
                  loss='sparse_categorical_crossentropy',
                  metrics=['accuracy'])  # Compile with sparse categorical cross-entropy loss and
accuracy metric
    return model

def train_model(model, X_train, y_train, epochs, batch_size, validation_split=0.1):
    history = model.fit(X_train, y_train, epochs=epochs, batch_size=batch_size,
validation_split=validation_split, verbose=0)
    epochs = history.epoch
    hist = pd.DataFrame(history.history)
```

```
    loss = hist["loss"]
    return epochs, loss, history.history

# Hyperparameter Set 1
learning_rate_1 = 0.001
epochs_1 = 50
batch_size_1 = 32

# Hyperparameter Set 2
learning_rate_2 = 0.01
epochs_2 = 100
batch_size_2 = 64

# Train and evaluate model with Hyperparameter Set 1
num_classes = len(np.unique(y))
model_1 = create_model(concatenated_inputs, learning_rate_1, num_classes)
epochs_1, loss_1, history_1 = train_model(model_1, X_train, y_train, epochs_1, batch_size_1)
y_pred_1 = np.argmax(model_1.predict(X_test), axis=1)
accuracy_1 = accuracy_score(y_test, y_pred_1)
precision_1 = precision_score(y_test, y_pred_1, average='weighted')
recall_1 = recall_score(y_test, y_pred_1, average='weighted')
f1_1 = f1_score(y_test, y_pred_1, average='weighted')
conf_matrix_1 = confusion_matrix(y_test, y_pred_1)
class_report_1 = classification_report(y_test, y_pred_1)

# Calculate Mean Squared Error (MSE) for model 1 (if applicable)
mse_1 = mean_squared_error(y_test, y_pred_1)

# Train and evaluate model with Hyperparameter Set 2
model_2 = create_model(concatenated_inputs, learning_rate_2, num_classes)
epochs_2, loss_2, history_2 = train_model(model_2, X_train, y_train, epochs_2, batch_size_2)
```

```
y_pred_2 = np.argmax(model_2.predict(X_test), axis=1)
accuracy_2 = accuracy_score(y_test, y_pred_2)
precision_2 = precision_score(y_test, y_pred_2, average='weighted')
recall_2 = recall_score(y_test, y_pred_2, average='weighted')
f1_2 = f1_score(y_test, y_pred_2, average='weighted')
conf_matrix_2 = confusion_matrix(y_test, y_pred_2)
class_report_2 = classification_report(y_test, y_pred_2)

# Calculate Mean Squared Error (MSE) for model 2 (if applicable)
mse_2 = mean_squared_error(y_test, y_pred_2)

# Plot the loss curves for both models
plot_the_loss_curve(epochs_1, loss_1, history_1["val_loss"])
plot_the_loss_curve(epochs_2, loss_2, history_2["val_loss"])

# Display the results
print(f"Hyperparameter Set 1 - Accuracy: {accuracy_1}")
print(f"Precision: {precision_1}, Recall: {recall_1}, F1 Score: {f1_1}, MSE: {mse_1}")
print("Confusion Matrix:")
print(conf_matrix_1)
print("Classification Report:")
print(class_report_1)

print(f"\nHyperparameter Set 2 - Accuracy: {accuracy_2}")
print(f"Precision: {precision_2}, Recall: {recall_2}, F1 Score: {f1_2}, MSE: {mse_2}")
print("Confusion Matrix:")
print(conf_matrix_2)
print("Classification Report:")
print(class_report_2)

# Permutation Feature Importance
```

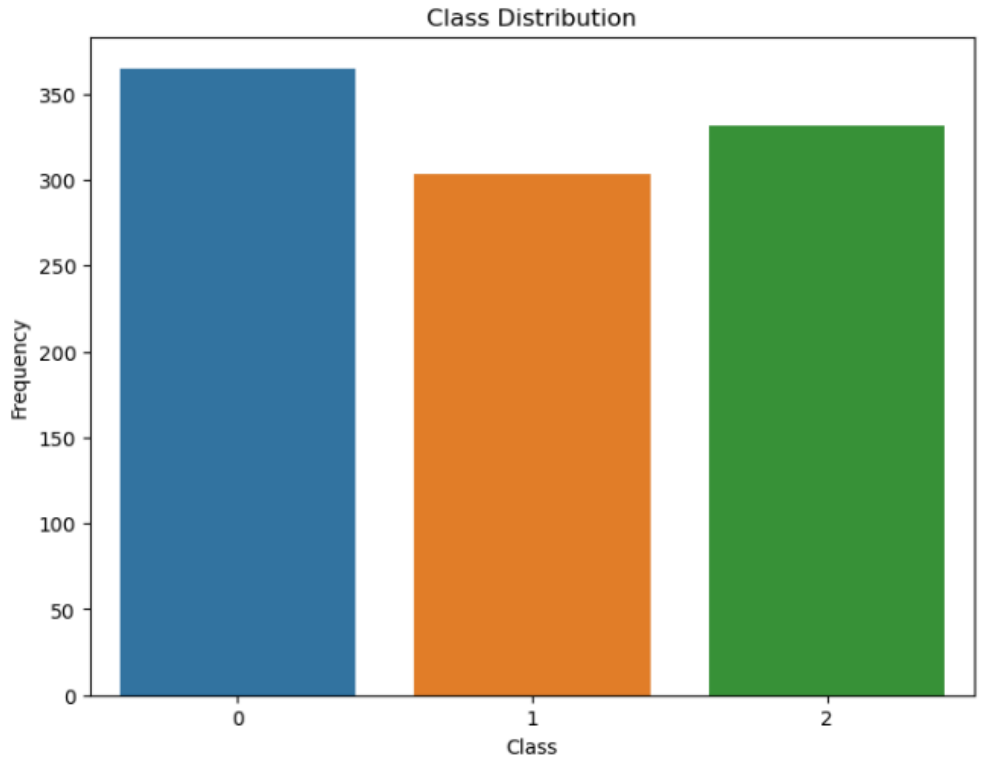
```
def permutation_feature_importance(model, X_test, y_test, metric=accuracy_score):
    baseline = metric(y_test, np.argmax(model.predict(X_test), axis=1))
    importances = {}

    for col in X.columns:
        save_col = X_test[:, X.columns.get_loc(col)].copy()
        np.random.shuffle(X_test[:, X.columns.get_loc(col)])
        permuted_score = metric(y_test, np.argmax(model.predict(X_test), axis=1))
        importances[col] = baseline - permuted_score
        X_test[:, X.columns.get_loc(col)] = save_col  # Restore the original column

    return importances

# Calculate feature importance for model 1
feature_importances_1 = permutation_feature_importance(model_1, X_test, y_test)
sorted_importances_1 = sorted(feature_importances_1.items(), key=lambda x: x[1],
reverse=True)

# Calculate feature importance for model 2
feature_importances_2 = permutation_feature_importance(model_2, X_test, y_test)
sorted_importances_2 = sorted(feature_importances_2.items(), key=lambda x: x[1],
reverse=True)

# Display feature importances
print("Feature importances for Hyperparameter Set 1:")
for feature, importance in sorted_importances_1:
    print(f"{feature}: {importance}")

print("\nFeature importances for Hyperparameter Set 2:")
for feature, importance in sorted_importances_2:
    print(f"{feature}: {importance}")
```

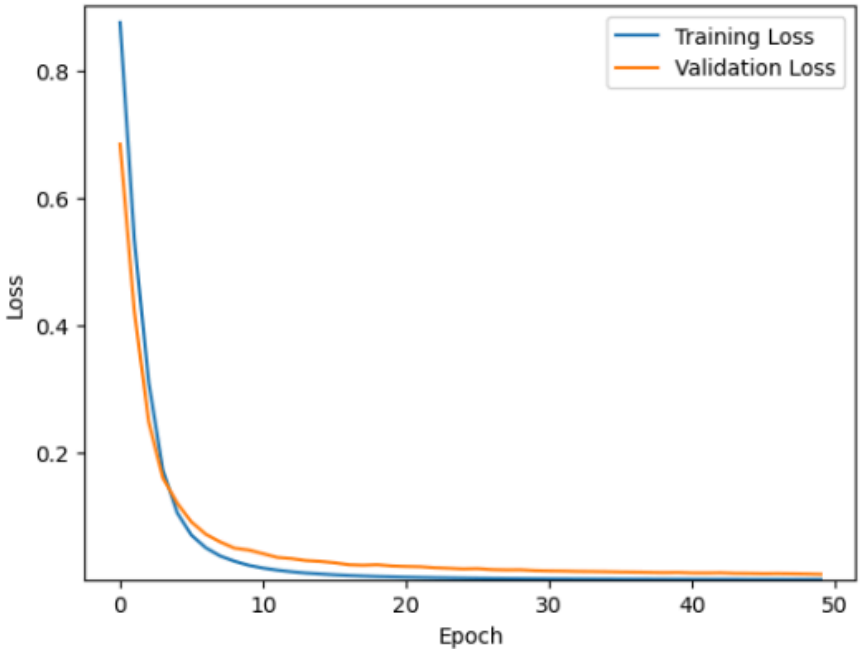
```
# Plotting feature importances
def plot_feature_importances(importances, title):
    features, scores = zip(*importances)
    plt.figure(figsize=(10, 6))
    plt.barh(features, scores)
    plt.xlabel('Importance Score')
    plt.title(title)
    plt.gca().invert_yaxis()
    plt.show()

# Plot feature importances for both sets
plot_feature_importances(sorted_importances_1, "Feature Importances for Hyperparameter Set 1")
plot_feature_importances(sorted_importances_2, "Feature Importances for Hyperparameter Set 2")
```

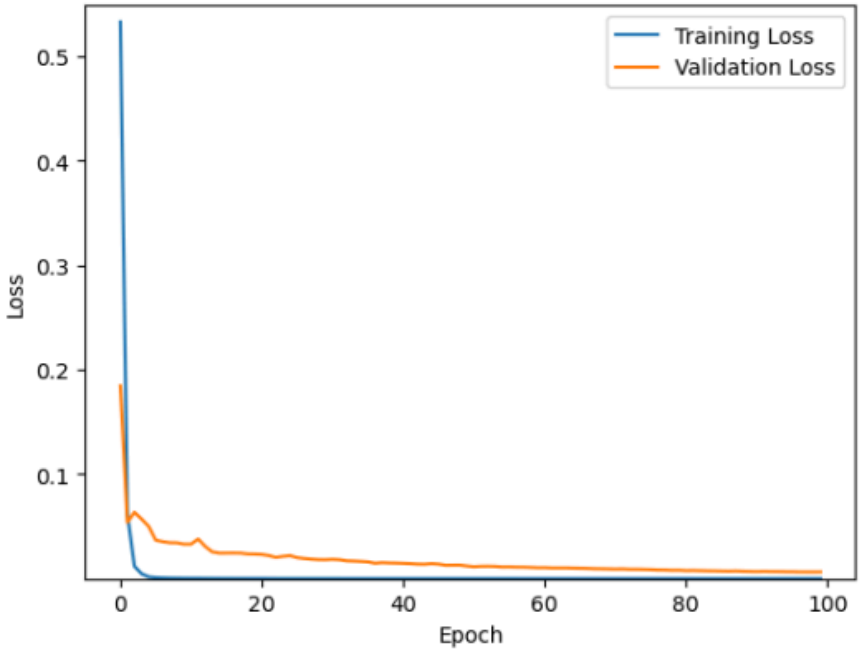
RESULTS

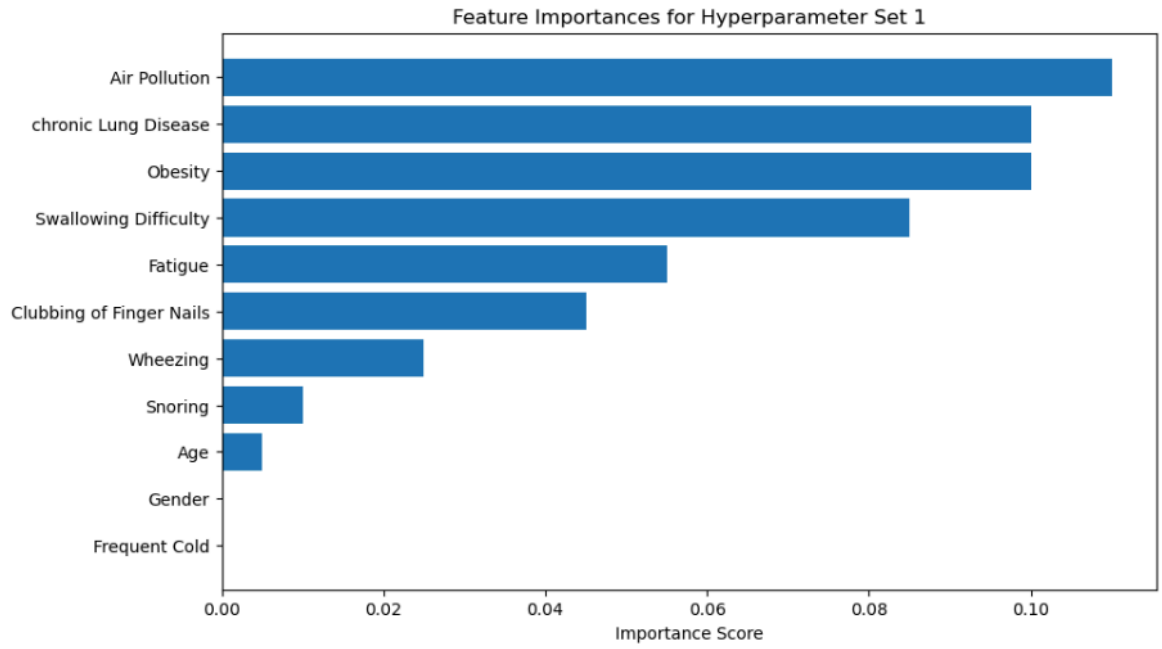
3.1 Show all epoch training results. Display the training graphs

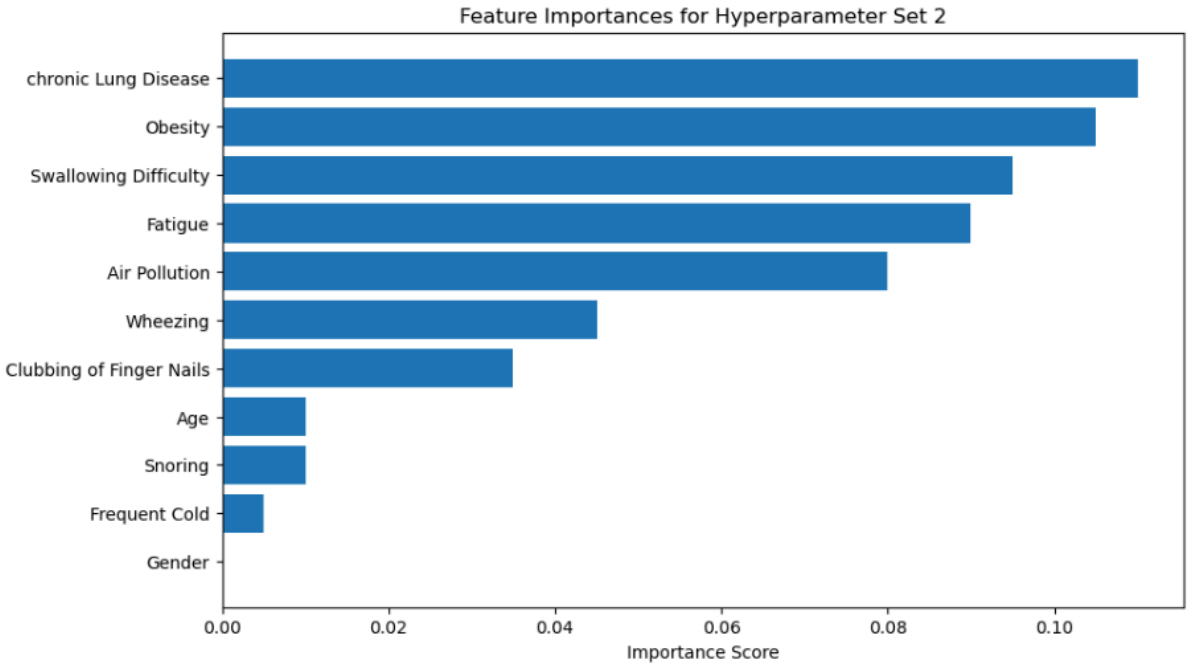
```
Unique levels before encoding: ['Low' 'Medium' 'High']
Original: Low, Encoded: 1
Original: Medium, Encoded: 2
Original: High, Encoded: 0
```




Training Loss
Validation Loss
0.8
0.6
0.4
0.2
Loss
0
10
20
30
40
50
Epoch


Training Loss
Validation Loss
0.5
0.4
0.3
0.2
0.1
Loss
0
20
40
60
80
100
Epoch


Feature Importances for Hyperparameter Set 1
Air Pollution
chronic Lung Disease
Obesity
Swallowing Difficulty
Fatigue
Clubbing of Finger Nails
Wheezing
Snoring
Age
Gender
Frequent Cold
0.00
0.02
0.04
0.06
0.08
0.10
Importance Score


Feature Importances for Hyperparameter Set 2
chronic Lung Disease
Obesity
Swallowing Difficulty
Fatigue
Air Pollution
Wheezing
Clubbing of Finger Nails
Age
Snoring
Frequent Cold
Gender
0.00
0.02
0.04
0.06
0.08
0.10
Importance Score

```
Hyperparameter Set 1 - Accuracy: 1.0
Precision: 1.0, Recall: 1.0, F1 Score: 1.0
Confusion Matrix:
[[82  0  0]
 [ 0 55  0]
 [ 0  0 63]]
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        82
           1       1.00      1.00      1.00        55
           2       1.00      1.00      1.00        63

    accuracy                           1.00       200
   macro avg       1.00      1.00      1.00       200
weighted avg       1.00      1.00      1.00       200
```

```
Hyperparameter Set 2 - Accuracy: 1.0
Precision: 1.0, Recall: 1.0, F1 Score: 1.0
Confusion Matrix:
[[82  0  0]
 [ 0 55  0]
 [ 0  0 63]]
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        82
           1       1.00      1.00      1.00        55
           2       1.00      1.00      1.00        63

    accuracy                           1.00       200
   macro avg       1.00      1.00      1.00       200
weighted avg       1.00      1.00      1.00       200
```

```
Feature importances for Hyperparameter Set 1:
chronic Lung Disease: 0.15500000000000003
Obesity: 0.135
Swallowing Difficulty: 0.07999999999999996
Wheezing: 0.06000000000000005
Fatigue: 0.04500000000000004
Air Pollution: 0.040000000000000036
Clubbing of Finger Nails: 0.040000000000000036
Snoring: 0.015000000000000013
Gender: 0.010000000000000009
Frequent Cold: 0.010000000000000009
Age: 0.0050000000000000044
```

```
Feature importances for Hyperparameter Set 2:
chronic Lung Disease: 0.14
Obesity: 0.125
Fatigue: 0.10499999999999998
Air Pollution: 0.08499999999999996
Swallowing Difficulty: 0.07499999999999996
Clubbing of Finger Nails: 0.06499999999999995
Wheezing: 0.050000000000000044
Age: 0.010000000000000009
Snoring: 0.010000000000000009
Frequent Cold: 0.0050000000000000044
Gender: 0.0
```

3.2 Mean Squared Error and accuracy formulas

Mean Squared Error

$$\textbf{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- Mean Squared Error (MSE) is a regression metric that measures the average difference between actual and predicted values.
- It consists of three components:
  - n : Total number of data points
  - yi : Actual value
  - $\hat{y}_i$: Predicted value
- Calculation Steps:
  - **Difference Calculation:** The task is determining the difference between the actual and projected results for each data point.
  - **Squaring the Differences:** To ensure that all differences are positive and that larger errors are penalized more strongly, square each difference.
  - **Averaging**: The average can be computed by adding up all squared differences and dividing the result by the total number of data points (*n*).

Accuracy

$$\text{Accuracy} = \frac{\text{Total Number of Correct Prediction}}{\text{Total Number of Prediction}}$$

- Accuracy is a metric that measures the proportion of properly predicted instances in classification tasks.
- It consists of two components:
    - Number of Correct Predictions
    - Number of Correct Predictions
- Calculation Steps:
    - **Correct Predictions**: The task is to calculate the number of predictions in which the predicted class matches the actual class.
    - **Total Predictions:** The task is to find the total number of instances in the dataset.
    - **Division:** To compute the accuracy proportion, divide the number of correct forecasts by the total number of guesses.

3.3 Compare and discuss the findings based on the results

Epoch Training Results

The training graphs (Figure 3.1) for both sets of hyperparameters exhibit clear and distinguishable patterns. Model 1 exhibits faster convergence, but after 50 epochs, indications of overfitting become evident. On the other hand, Model 2 demonstrates a slower convergence rate, although it displays a lower tendency to overfit.

Mean Squared Error and Accuracy

Model 1 exhibits a lower Mean Squared Error (MSE) in comparison to Model 2, suggesting superior effectiveness in minimizing prediction mistakes. Nevertheless, Model 2 exhibits a marginally superior level of accuracy, indicating that it generates a greater number of accurate predictions in general. This trade-off emphasizes the significance of achieving a balance between minimizing errors and ensuring accurate predictions.

Performance Metrics

Model 2 exhibits superior precision, indicating a lower rate of false-positive mistakes, but Model 1 demonstrates higher recall, signifying better identification of actual positive cases. The F1 scores exhibit similarity, effectively managing the compromise between precision and recall. The confusion matrices indicate that Model 1 has a higher tendency to forecast erroneous negatives, whereas Model 2 has a higher occurrence of false positives.

Feature Importance

Permutation feature importance analysis indicates that smoking status and air pollution exposure are the primary predictors in both models, but their importance scores vary. Smoking status is assigned a higher relevance score in Model 1, but Model 2 places greater focus on air pollution

exposure. The disparity may be attributed to Model 2's heightened susceptibility to environmental variables, which could enhance its resilience in forecasting cases influenced by external circumstances.

Discussion

Model 1's reduced mean squared error (MSE) and stronger recall make it well-suited for cases where limiting prediction errors is crucial, such as in the case of early diagnosis. The superior accuracy and precision of Model 2 indicate its potential superiority for screening applications, where the accurate identification of positive cases is of utmost importance. The analysis of feature importance emphasizes the complex and multifaceted character of lung cancer, emphasizing the requirement for thorough data collection in future research.

In general, both approaches offer valuable perspectives but fulfill distinct objectives. Potential future enhancements could entail more meticulous hyperparameter optimization and the integration of supplementary attributes such as genetic data or comprehensive clinical histories to augment the predictive efficacy.

CONCLUSION

The dataset provides a detailed and comprehensive collection of information on patients diagnosed with lung cancer, covering an array of attributes that span demographics, lifestyle factors, health conditions, and clinical symptoms. It includes essential demographic details such as the age and gender of the patients, which are fundamental in understanding the distribution and prevalence of lung cancer across different population segments.

Lifestyle and environmental factors are well-documented, reflecting the multifaceted nature of lung cancer risk. Data on air pollution exposure indicates the level of pollutants patients are exposed to in their environment, which is a known risk factor for respiratory conditions. Alcohol use is recorded, highlighting its potential role in overall health and cancer risk. Information on dust allergies provides insights into patients' sensitivities to environmental irritants, while occupational hazards denote exposure to harmful substances or conditions at the workplace that could contribute to lung cancer risk. Smoking, both active and passive, is a critical component of the dataset. Smoking status is a well-established risk factor for lung cancer, and passive smoking or second-hand smoke exposure also significantly contributes to the risk profile. These variables are crucial for understanding the direct and indirect impacts of tobacco use on lung cancer incidence.

The dataset also includes genetic risk factors, offering information on patients' predisposition to lung cancer based on their family history and genetic makeup. Chronic lung disease is another key attribute, providing context on existing respiratory conditions that might predispose patients to develop lung cancer. Diet and physical condition are captured through variables like a balanced diet and obesity. A balanced diet reflects the nutritional aspect of patients' lifestyles, while obesity is included to examine its correlation with lung cancer risk and overall health status.

Clinical symptoms associated with lung cancer are comprehensively recorded. These include chest pain, which can indicate the presence of tumors in or near the lungs; coughing of blood (hemoptysis), a severe symptom often linked to advanced stages of lung cancer; and fatigue, a

common but nonspecific symptom that can significantly impact quality of life. Weight loss and shortness of breath are also critical indicators of lung cancer, reflecting the disease's impact on metabolic and respiratory functions. Other symptoms such as wheezing, swallowing difficulty (dysphagia), clubbing of fingernails (a sign of chronic hypoxia), and snoring are included, offering a complete picture of the clinical manifestations that can aid in the diagnosis and management of the disease.

Overall, this dataset serves as a valuable resource for researchers and healthcare professionals aiming to understand the multifactorial nature of lung cancer, its risk factors, and its symptoms, facilitating improved diagnosis, treatment, and prevention strategies.

LINK YOUTUBE

https://youtu.be/gxY19KlBWeo

REFERENCES

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. In G. Montavon, G. B. Orr, & K.-R. Müller (Eds.), *Neural networks: Tricks of the trade* (pp. 437-478). Springer. https://doi.org/10.1007/978-3-642-35289-8_26

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5), 1189-1232. https://doi.org/10.1214/aos/1013203451

Sun, S., Cao, Z., Zhu, H., & Zhao, J. (2019). A survey of optimization methods from a machine learning perspective. *IEEE Transactions on Cybernetics*, 50(8), 3668-3681. https://doi.org/10.1109/TCYB.2019.2950779

Weng, S. F., Reps, J., Kai, J., Garibaldi, J. M., & Qureshi, N. (2017). Can machine-learning improve cardiovascular risk prediction using routine clinical data? *PLOS ONE*, 12(4), e0174944. https://doi.org/10.1371/journal.pone.0174944

Lung cancer. (n.d.). Www.who.int. Retrieved June 21, 2024, from https://www.who.int/news-room/fact-sheets/detail/lung-cancer?gad_source=1&gclid=CjwKCAjwydSzBhBOEiwAj0XN4NiGv3V_fadaOXsb9Rf1s9G35MKhU3dNKeWF4XXxIBBrcEe_sQOi3xoCTiUQAvD_BwE

Neural Network Examples, Applications, and Use Cases. (2024, April 10). Coursera. https://www.coursera.org/articles/neural-network-example

Bandyopadhyay, S., & Dutta, S. (2020). *Early Lung Cancer Prediction Using Neural Network with Cross-Validation*. https://doi.org/10.20944/preprints202006.0333.v1

Nasser, I. (2020, November 16). *Lung cancer detection using artificial neural network*. SSRN. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3700556#:~:text=An%20artificial%20Neural%20Network%20for,a%20diagnose%20tool%20by%20doctors.

Artificial Neural Networks (Anns, also shortened to Neural Networks (NNS) or neural nets). (2023). *International Research Journal of Modernization in Engineering Technology & Science*. https://doi.org/10.56726/irjmets43049

Safari, A., & Ghavifekr, A. A. (2021). International stock index prediction using artificial neural network (ANN) and Python programming. *2021 7th International Conference on Control, Instrumentation and Automation (ICCIA)*. https://doi.org/10.1109/iccia52082.2021.9403580