Lung Cancer Prediction

- **Kanjani Vanshkumar**

1. **Dataset Description: -**

The effectiveness of cancer prediction system helps the people to know their cancer risk with low cost and it also helps the people to take the appropriate decision based on their cancer risk status. The data is collected from the website online lung cancer prediction system .

This dataset contains 1236 rows and 16 attributes.

Content: -

1. Gender: M(male), F(female)
2. Age: Age of the patient
3. Smoking: YES=2 , NO=1.
4. Yellow fingers: YES=2 , NO=1.
5. Anxiety: YES=2 , NO=1.
6. Peer_pressure: YES=2 , NO=1.
7. Chronic Disease: YES=2 , NO=1.
8. Fatigue: YES=2 , NO=1.
9. Allergy: YES=2 , NO=1.
10. Wheezing: YES=2 , NO=1.
11. Alcohol: YES=2 , NO=1.
12. Coughing: YES=2 , NO=1.
13. Shortness of Breath: YES=2 , NO=1.
14. Swallowing Difficulty: YES=2 , NO=1.
15. Chest pain: YES=2 , NO=1.
16. Lung Cancer: YES , NO.

2. **Data Pre-processing: -**

2.1. Data Import: -

```
[3] df = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/SURVEY_LUNG_CANCER_DATASET_PRED.csv")
```

```
df
```

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL CONSUMING	COUGHING	SHORTNESS OF BREATH	SWALLOWING DIFFICULTY	CHEST PAIN	LUNG_CANCER
0	M	69	1	2	2	1	1	2	1	2	2	2	2	2	2	YES
1	M	74	2	1	1	1	2	2	2	1	1	1	2	2	2	YES
2	F	59	1	1	1	2	1	2	1	2	1	2	2	1	2	NO
3	M	63	2	2	2	1	1	1	1	1	2	1	1	2	2	NO
4	F	63	1	2	1	1	1	1	1	2	1	2	2	1	1	NO
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1231	F	56	1	1	1	2	2	2	1	1	2	2	2	2	1	YES
1232	M	70	2	1	1	1	1	2	2	2	2	2	2	1	2	YES
1233	M	58	2	1	1	1	1	1	2	2	2	2	1	1	2	YES
1234	M	67	2	1	2	1	1	2	2	1	2	2	2	1	2	YES
1235	M	62	1	1	1	2	1	2	2	2	2	1	1	2	1	YES

1236 rows × 16 columns

2.2. Missing Values: - No missing values.

```
df.isnull().sum() # total null values
```

```
GENDER                    0
AGE                       0
SMOKING                   0
YELLOW_FINGERS            0
ANXIETY                   0
PEER_PRESSURE             0
CHRONIC DISEASE           0
FATIGUE                   0
ALLERGY                   0
WHEEZING                  0
ALCOHOL CONSUMING         0
COUGHING                  0
SHORTNESS OF BREATH       0
SWALLOWING DIFFICULTY     0
CHEST PAIN                0
LUNG_CANCER               0
dtype: int64
```

2.2. Data Encoding: - encoded two columns i.e. GENDER,LUNG_CANCER using labelencoder .

```
[9]  #GENDER                # Converting String to float
     from sklearn.preprocessing import LabelEncoder
     labelencoder = LabelEncoder()
     X = df.iloc[:,0].values
     df.iloc[:,0] = labelencoder.fit_transform(X)
     X = X.reshape(-1,1)
```

```
[10] #LUNG CANCER               # Converting String to float
     from sklearn.preprocessing import LabelEncoder
     labelencoder = LabelEncoder()
     X = df.iloc[:,15].values
     df.iloc[:,15] = labelencoder.fit_transform(X)
     X = X.reshape(-1,1)
```

2.4. Train Test Split: -

1. PCA: -

```
[16] from sklearn.model_selection import train_test_split
     X_train_pca, X_test_pca, Y_train_pca, Y_test_pca = train_test_split(X_new, y, test_size = 0.25)

     print(X_train_pca.shape)
     print(Y_train_pca.shape)
     print(X_test_pca.shape)
     print(Y_test_pca.shape)

     (927, 1)
     (927,)
     (309, 1)
     (309,)
```

2. **MinMax: -**

```
from sklearn.model_selection import train_test_split
X_train_mms, X_test_mms, Y_train_mms, Y_test_mms = train_test_split(X, y, test_size = 0.25)

print(X_train_mms.shape)
print(Y_train_mms.shape)
print(X_test_mms.shape)
print(Y_test_mms.shape)
```

```
(927, 15)
(927,)
(309, 15)
(309,)
```

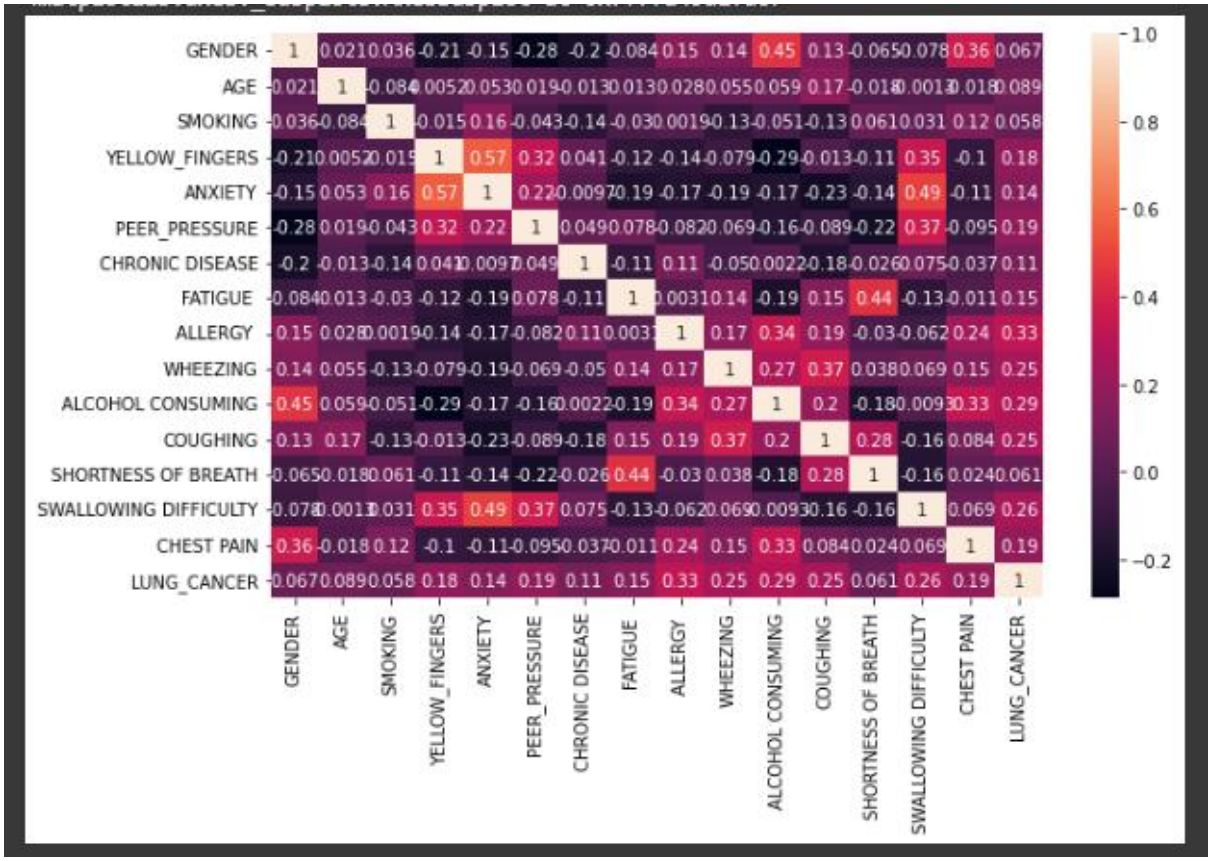
2.5. Scaling: -

1. **Standard** – It removes the mean and scales each feature/variable to unit variance.

2. **PCA** - Principal component analysis is a technique for reducing the dimensionality of such datasets, increasing interpretability but at the same time minimizing information loss.

3. **MinMax** – It transform features by scaling each feature to a given range.

Corelation Diagram: -



3. Model Selection: -

1) Importing - Standard, PCA and MinMax Scalar.
2) PCA – Logistic Regression, Decision Tree, Random Forest, SVM Classifier, KNN, Naive Bayes.
3) MinMax – Logistic Regression, Decision Tree, Random Forest, SVM Classifier, KNN, Naive Bayes.

4. Result and Discussion: -

Technology used: Python;

Libraries: numpy, pandas, sklearn, matplotlib.pyplot, seaborn, sklearn.preprocessing, sklearn.metrics.

4.1. Output Tables: -

Table 1: Model comparison for **Lung cancer** prediction using **PCA**.

Models	Prediction	Accuracy	Precision	Recall	F1 - Score
Logistic Regression	HAVING_CANCER NOT_HAVING_CANCER	0.87	0.00 0.87	0.00 1.00	0.00 0.93
Decision Tree	HAVING_CANCER NOT_HAVING_CANCER	0.99	0.95 1.00	1.00 0.99	0.98 1.00
KNN	HAVING_CANCER NOT_HAVING_CANCER	0.88	0.60 0.89	0.19 0.98	0.29 0.93
SVM Classifier	HAVING_CANCER NOT_HAVING_CANCER	0.87	0.00 0.87	0.00 1.00	0.00 0.93
Random forest	HAVING_CANCER NOT_HAVING_CANCER	0.97	0.88 0.99	0.91 0.98	0.89 0.98
Naive Bayes	HAVING_CANCER NOT_HAVING_CANCER	0.87	0.00 0.87	0.00 1.00	0.00 0.93

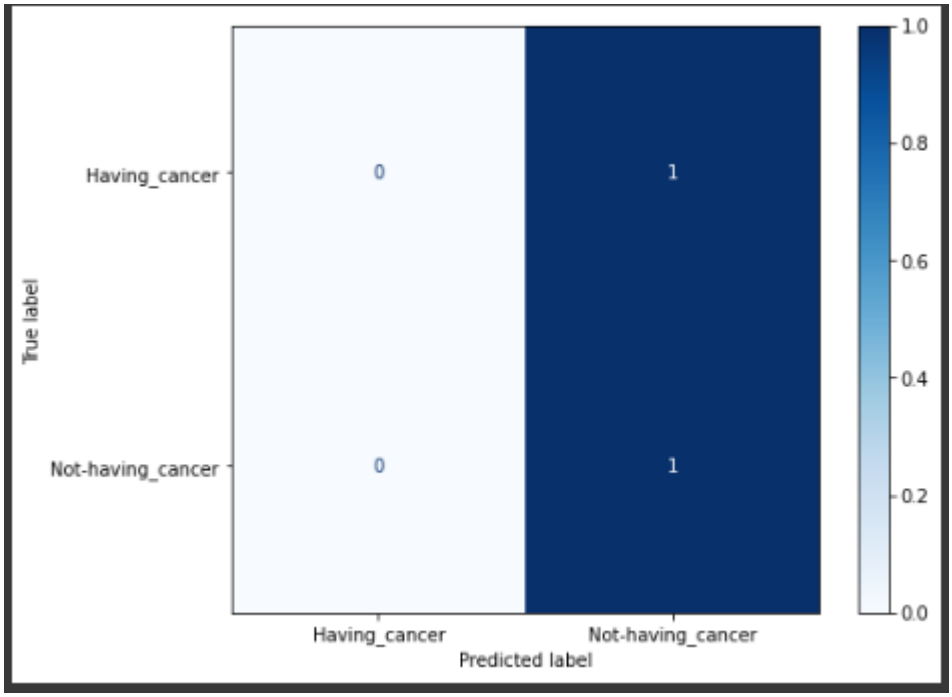
Table 2: Model comparison for **Lung cancer** prediction using **MinMax**.

Models	Prediction	Accuracy	Precision	Recall	F1 - Score
Logistic Regression	HAVING_CANCER NOT_HAVING_CANCER	0.94	0.78 0.95	0.64 0.98	0.70 0.96
Decision Tree	HAVING_CANCER NOT_HAVING_CANCER	0.99	0.98 1.00	1.00 1.00	0.99 1.00
KNN	HAVING_CANCER NOT_HAVING_CANCER	0.94	0.74 0.96	0.70 0.97	0.72 0.97
SVM Classifier	HAVING_CANCER NOT_HAVING_CANCER	0.96	0.89 0.97	0.80 0.98	0.84 0.98
Random Forest	HAVING_CANCER NOT_HAVING_CANCER	0.99	1.00 0.99	0.95 1.00	0.97 1.00
Naive Bayes	HAVING_CANCER NOT_HAVING_CANCER	0.89	0.56 0.93	0.51 0.94	0.53 0.94

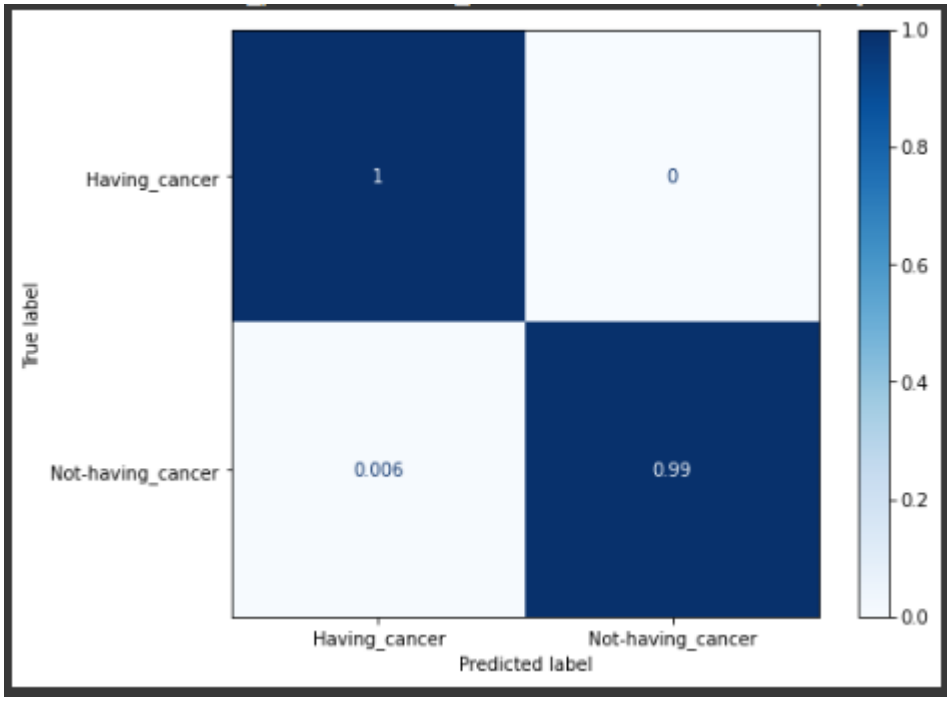
4.2. Confusion Matrix: -

1. PCA: -

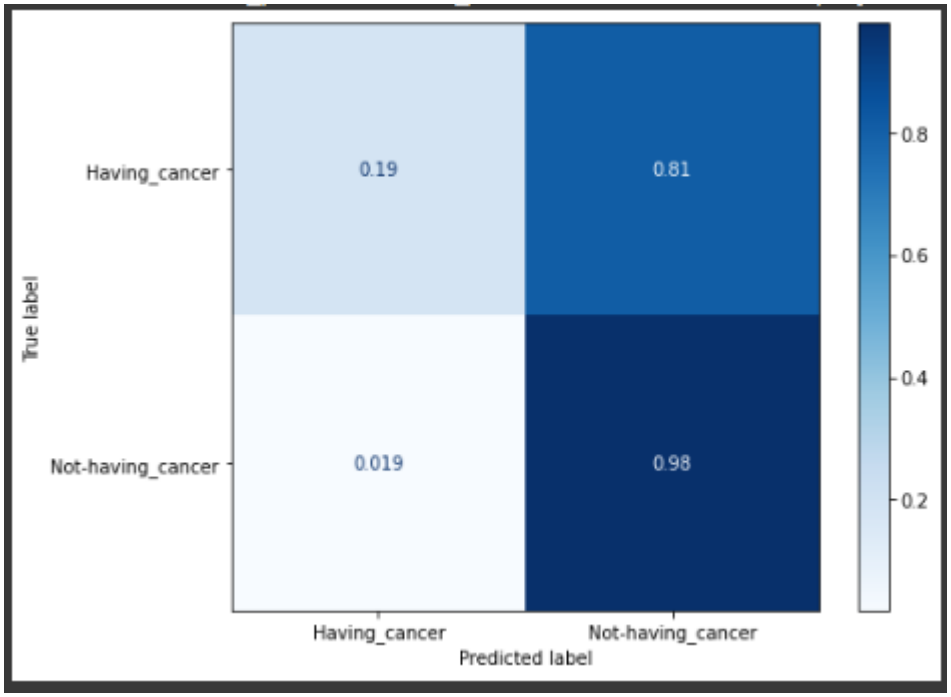
a) Logistic Regression – PCA



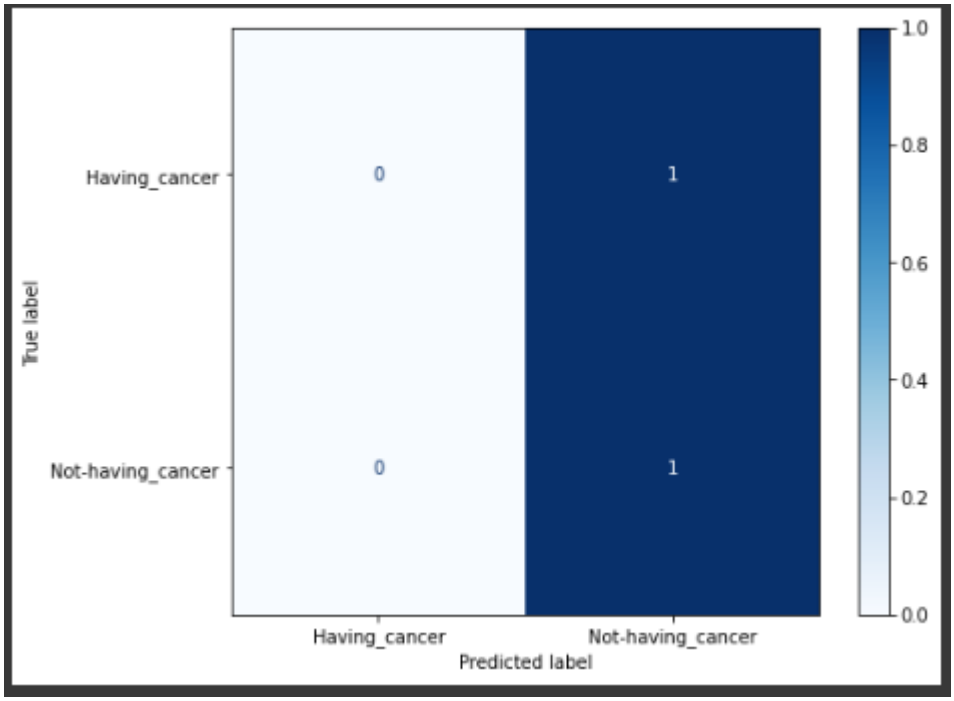
b) Decision Tree – PCA


Having_cancer
Not-having_cancer
True label
1
0
0.006
0.99
Having_cancer
Not-having_cancer
Predicted label
1.0
0.8
0.6
0.4
0.2
0.0

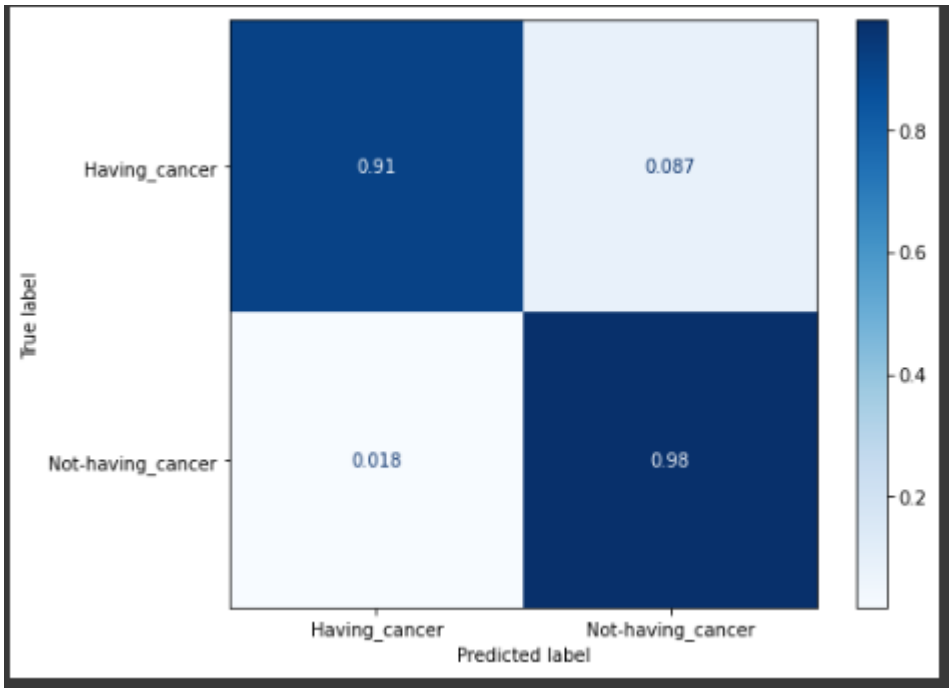
c) KNN – PCA



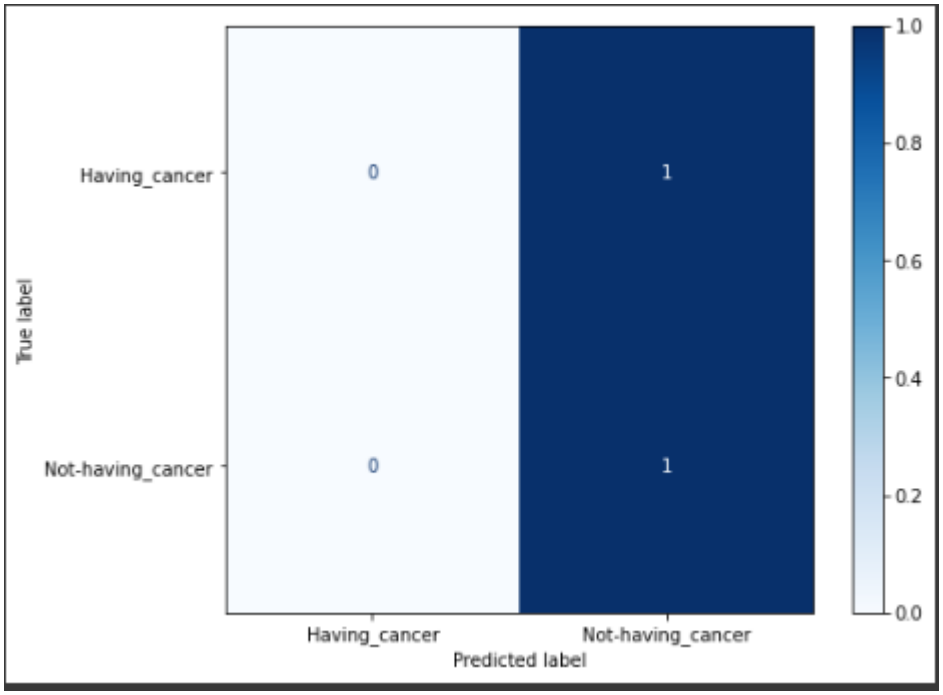
d) SVM Classifier – PCA



e) Random Forest – PCA

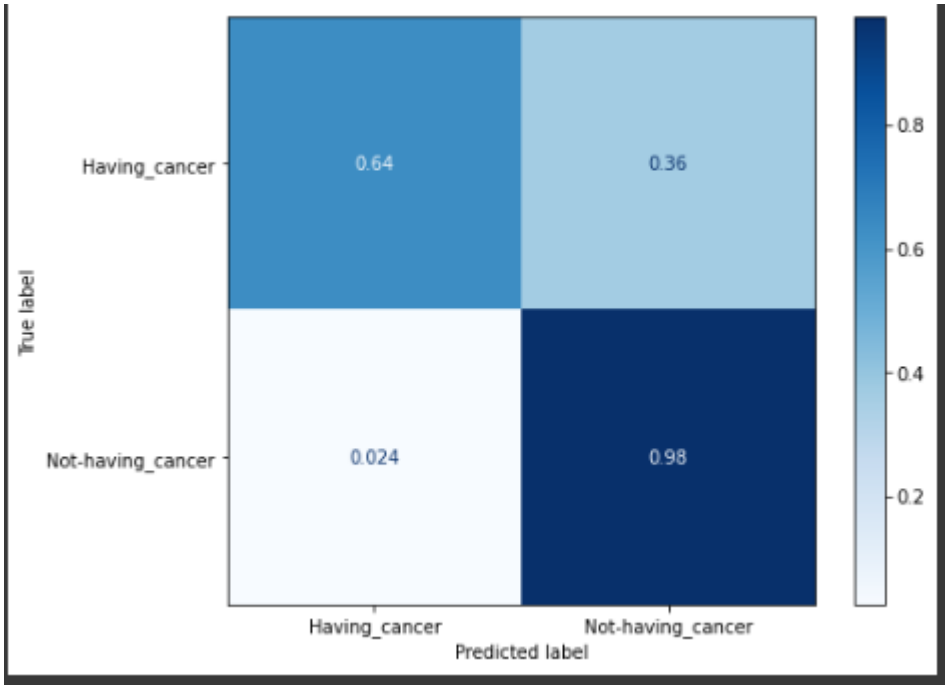


f) Naive Bayes – PCA

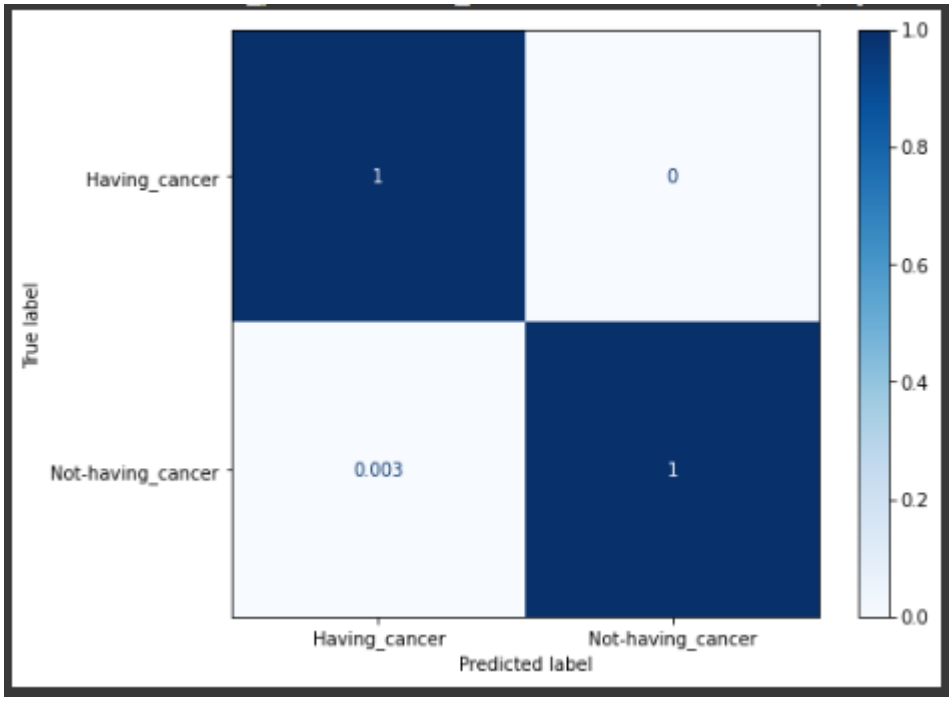


2. MinMax: -

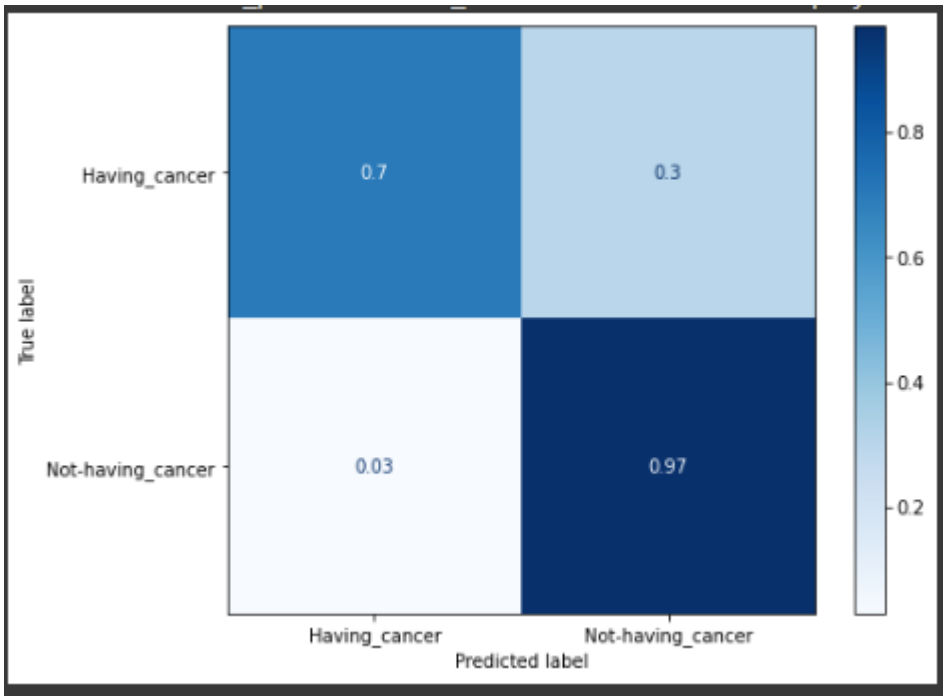
a) Logistic Regression – MinMax



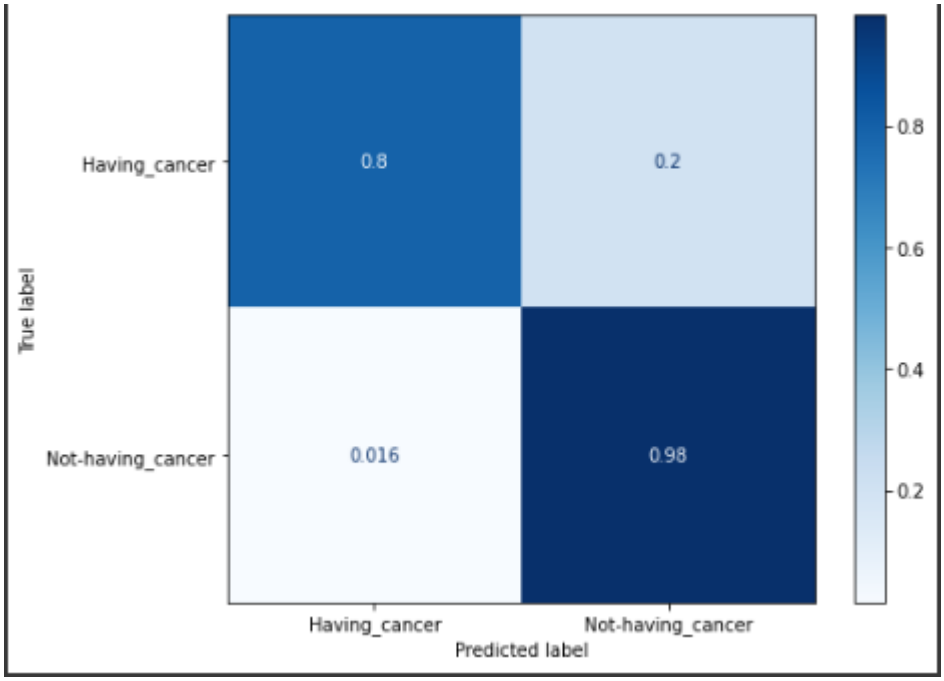
b) Decision Tree – MinMax


Having_cancer
Not-having_cancer
True label
1
0
0.003
1
Having_cancer
Not-having_cancer
Predicted label
1.0
0.8
0.6
0.4
0.2
0.0

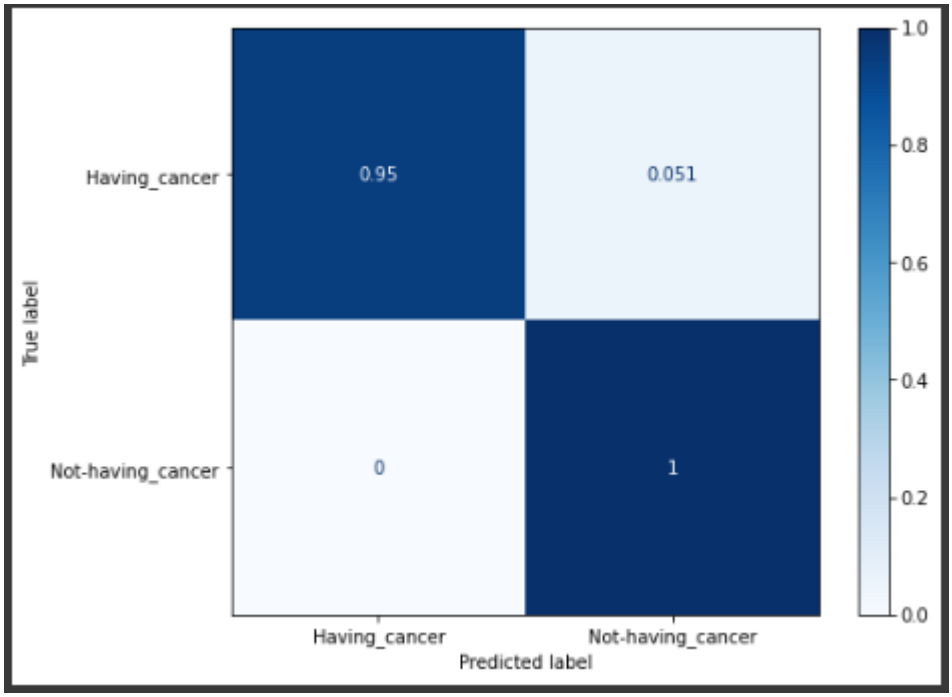
c) KNN– MinMax


Having_cancer
Not-having_cancer
True label
0.7
0.3
0.03
0.97
Having_cancer
Not-having_cancer
Predicted label
0.8
0.6
0.4
0.2

d) SVM Classifier – MinMax



e) Random Forest – MinMax



f) Naive Bayes – MinMax