LUNG CANCER SCREENING MODELS REPORT

GOITOM HADGU W2064676

JULY 1, 2024
UNIVERSITY OF WESTMINSTER
7BUIS008W Data Mining & Machine Learning
Module leader Dr. Mahmoud Aldraimli

TASK I

Variable Name	Retain or Drop	Brief justification for retention or dropping
Patient ID	Drop	Personal information that is irrelevant to our desired model.
Genomic Sex	Retain	According to the research (May et al., 2023), The influence of biological differences, including the impact of sex hormones and variations in immune response, is high. lung cancer shows sex-specific trends
Age	Retain	Age has a strong relation with lung cancer. According to a study, the chance of developing lung cancer increases with age (Eldridge, 2023)
Blood type	Drop	According to a report, the occurrence of lung cancer was found to be independent of blood type. (Yang et al., 2022)
Siblings	Drop	The risk of lung cancer is unlikely to be directly affected by the number of siblings.
Year of Birth	Drop	This feature is the same age, so it should be dropped.
Month of birth	Drop	No relevant relation with Lung cancer.
Adaption	Drop	Though it is recommended to know family history, It does not directly relate to the risk of lung cancer.
Pregnancy	Drop	The number of missing values is high.
Parent Alive	Drop	The status of parents does not directly impact the risk of lung cancer in patients.

Smoking Status	Retain	One of the most significant risk factors for developing lung cancer is smoking.
Daily Cigarettes	Retain	Highly correlated with lung cancer since smoking is the prime risk factor for cancer.
Yellow Skin	Retain	According to research by (the American Cancer Society, 2019), Unusual skin changes can be related to symptoms of lung cancer progression or side effects of lung cancer.
Anxiety	Retain	Patients with lung cancer are significantly more likely to have had a major stressful life event within the preceding 5 years (Jafri et al., 2017)
Peer Pressure	Drop	Peer pressure can lead to an increase in tobacco smoking, but it's hard to consider it a significant risk factor.
COPD Diagnosis	Retain	Researchers indicate that COPD, particularly the emphysema-dominant type, independently poses a prognostic risk for lung cancer. (Houghton, 2013)
Fatigue	Retain	Studies suggest lung cancer Fatigue is one of the most common and debilitating symptoms experienced by people with lung cancer (Carnio, Di Stefano and Novello, 2016)
Allergy	Drop	There is no strong evidence linking general allergies with the risk of lung cancer.

Wheezing	Retain	It is a Symptom associated with respiratory issues, including lung cancer (National Health Service, 2022)
Alcohol Consumption	Retain	Studies suggest that alcohol may independently contribute to the development of lung cancer, particularly in individuals with a genetic predisposition for the disease (Thompson et al., 2020).
Coughing	Retain	It is a significant risk factor and common symptom of lung cancer. (NHS, 2019)
Shortness of Breath	Retain	Shortness of breath continues to be a distressing symptom associated with lung cancer (John Hopkins Medicine, 2019)
Swallowing Difficulty	Retain	A symptom of advanced lung cancer.
Chest Pain	Retain	As a significant symptom, this could indicate potential lung issues, possibly including cancer.
Lung Cancer	Retain	It is our target variable.

TASK II DATA UNDERSTANDING

```
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   GENOMIC SEX            1101 non-null   object
 1   AGE                    1117 non-null   object
 2   SMOKING_STATUS         1120 non-null   int64
 3   DAILY_CIGARETTES       617 non-null    object
 4   YELLOW_SKIN            1120 non-null   int64
 5   ANXIETY                1120 non-null   int64
 6   COPD_DIAGNOSES         1102 non-null   float64
 7   FATIGUE                1115 non-null   float64
 8   WHEEZING               1110 non-null   float64
 9   ALCOHOL_CONSUMPTION    1120 non-null   int64
 10  COUGHING               1120 non-null   int64
 11  SHORTNESS_OF_BREATH    1116 non-null   float64
 12  SWALLOWING_DIFFICULTY  1120 non-null   int64
 13  CHEST_PAIN             1120 non-null   int64
 14  LUNG_CANCER            1117 non-null   object
dtypes: float64(4), int64(7), object(4)
memory usage: 131.4+ KB
```

Figure 1 Data type

	SMOKING_STATUS	YELLOW_SKIN	ANXIETY	COPD_DIAGNOSES	FATIGUE	WHEEZING	ALCOHOL_CONSUMPTION	COUGHING	SHORTNESS_OF_BREATH	SWALLOWING_DIFFICULTY	CHEST_PAIN
count	1120.000000	1120.000000	1120.000000	1102.000000	1115.000000	1110.000000	1120.000000	1120.000000	1116.000000	1120.000000	1120.000000
mean	1.550893	1.559821	1.491964	1.503630	1.634978	1.544144	1.550893	1.567857	1.624552	1.475000	1.551786
std	0.497625	0.496630	0.500159	0.500214	0.481652	0.498272	0.497625	0.495595	0.484455	0.499598	0.497533
min	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
25%	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
50%	2.000000	2.000000	1.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	1.000000	2.000000
75%	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000
max	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000

Figure 2 Description of the data

```
Genomic_Sex: Nominal
Age: Ratio
Smoking_Status: Nominal
Anxiety: Nominal
DAILY_CIGARETTE: Numeric
COPD_Diagnosis: Nominal
Fatigue: Nominal
Wheezing: Nominal
Yellow_skin: Nominal
Alcohol_Consumption: Ratio
Coughing: Nominal
Shortness_of_Breath: Nominal
Swallowing_Difficulty: Nominal
Chest_Pain: Nominal
Lung_Cancer: Nominal
```

Figure 3 Measurment scale

```
GENOMIC SEX              19
AGE                       3
SMOKING_STATUS            0
DAILY_CIGARETTES        503
YELLOW_SKIN               0
ANXIETY                   0
COPD_DIAGNOSES           18
FATIGUE                   5
WHEEZING                 10
ALCOHOL_CONSUMPTION       0
COUGHING                  0
SHORTNESS_OF_BREATH       4
SWALLOWING_DIFFICULTY     0
CHEST_PAIN                0
LUNG_CANCER               3
dtype: int64
```

Figure 4 Sum of missing/ null values

```
df.shape
(1120, 26)
```

Figure 4 dataframe shape

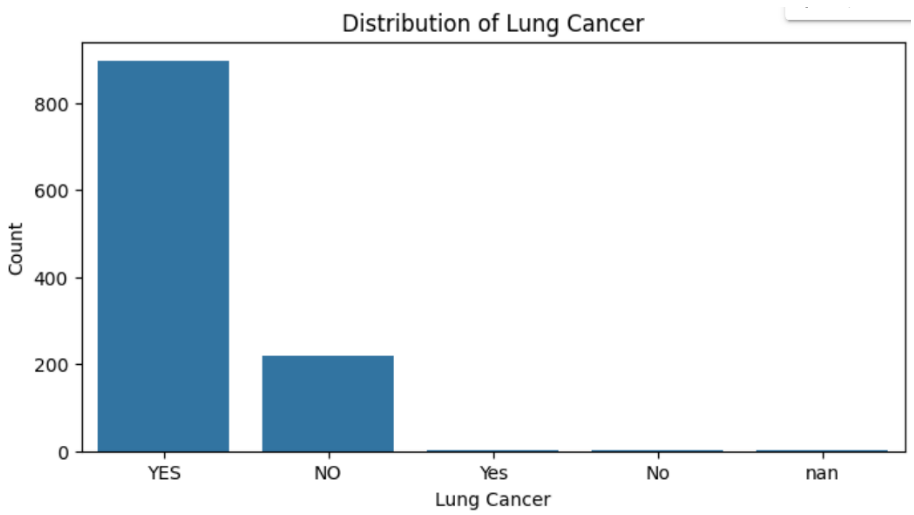


Figure 5 Distribution of target variable based on count

Task (3) –DATA PREPARATION: Cleaning and Transforming your data

Variable Name	Issue description	Proposed mitigation	Justification for used mitigation
Genomic Sex	-Technical feature name. -Missing values -Formatting issue. -Object Data type issue. -Categorical labels.	Rename using the rename function. Followed by a label encoding Male to 1 and Female to 0. Fill in the missing values using mode and change the data type to integer.	Renaming ensures consistency and clarity. Normalising formats removes inconsistency and eases analysis. Filling missing values with mode maintains the most common value in the dataset. Changing the data type to an integer ensures correct numeric operations.
AGE	-High scale. -Missing values. -Outliner ages. -Negative value -string values. -Object Data type.	By replacing the outlier age after capping it at 120 with the median age, filling the missing values with the median age, applying the absolute function, defining a text-to-numeric converter function, then changing the data type to integer. additionally by standardising it with min-max scaler.	To make our dataset suitable for processing by some statistical algorithms like linear regression SVM and KNN, the min-max scaler transforms the dataset for this. mapping outliers prevent unrealistic age values. The median is robust to outliers for filling in missing values. The absolute function

			ensures that all ages are positive. Text-to-numeric conversion ensures consistent data type. Standardising improves model performance and convergence.
DAILY CIGARETTE	-Missing value. -Outliners on the number of daily cigarettes. -Has String value. -Object data type. -High scale.	The missing value can be fixed by creating a logical relation (if it is a non-smoker 1, then the daily cigarette is 0) between smoking status and daily cigarettes, converting the text to numeric values and capping the number of daily cigarettes to 40 to remove the outliner, additionally, by standardising it min max scaler.	Logical relation leverages existing data for accuracy. Text-to-numeric conversion ensures consistent data type. Capping outliers prevents extreme values from skewing results. Standardising improves model performance and convergence. since our data doesn't follow Gaussian distribution, we used a min-max scaler to scale down.
COPD_DIAGNOSES	-Missing values. -Float data type. -Non binary label.	Handling the missing values with the most frequent value(mode) and ensuring suitable data type.	Improves the performance and interpretability of machine learning models. Numeric format maintains integrity and prevents mixed data type issues.
WHEEZING	-Missing value. -Float data type. -Non-binary labels.	Handling the missing values by replacing them with the most frequent value (mode) and converting the data type to integer. Using binary encoding for transformation.	Improves the performance and interpretability of machine learning models. Numeric format maintains integrity and prevents mixed data type issues.
FATIGUE	-Missing value. -The data type is float. -Non-binary Labels.	Handling the missing values with the most frequent value (mode) and changing the data type to integer. Binarising the values.	Mode is appropriate for categorical data. Correct data type. It ensures appropriate processing. Binarisation makes it

			suitable for our classification model.
SHORTNESS OF BREATH	-Missing value -The data type is float. -Non-binary labels.	Handling the missing values with the most frequent value(mode) and changing the data type to integer followed by binarising the label.	Mode is appropriate for categorical data. Correct data type ensures appropriate processing.
LUNG CANCER	-Formatting issue. -Object data type -Missing values. -Categorical Labels.	Standardise values by label encoding(Yes = 1, No = 0), drop the missing value, and change the data type to integer.	Label encoding ensures consistency in the binary representation of our feature. Since our feature is a target variable, the suitable way to handle the missing values is by dropping it. Correct data type ensures appropriate processing.
CHEST PAIN	-Non-binary labels.	Binary encoding.	Maintaining data in numeric format maintains integrity and prevents mixed data type issues.
COPD DIAGNOSIS	-Missing values. -Non-binary labels.	Handling the missing values with the most frequent value(mode) and Binary encoding.	Maintaining data in numeric format maintains integrity and prevents mixed data type issues.
YELLOW SKIN	-Non-binary labels.	Binary encoding.	Maintaining data in numeric format maintains integrity and prevents mixed data type issues.
ALCOHOL CONSUMPTION	-Non-binary labels.	Binary encoding.	Maintaining data in numeric format maintains integrity and prevents mixed data type issues.
COUGHING	-Non-binary labels.	Binary encoding.	Maintaining data in numeric format maintains integrity and prevents mixed data type issues.

PART B

```
[31] df.isnull().sum()

GENOMIC SEX            19
```

```
[39] df['GENDER'].unique()

array(['M', 'F', nan, 'MALE', 'FEMALE'], dtype=object)
```

Figure 6 Info of genomic sex/ Gender before implementation of changes.

Issues resolved:

- Changed feature name to GENDER.
- Filled missing values with mode.
- Standardized values (Male = 1, Female = 0).
- Changed data type to integer.

```
0       1
1       1
2       0
3       1
4       0
       ..
1115    0
1116    0
1117    0
1118    0
1119    0
Name: GENDER, Length: 1120, dtype: int64
```

Figure 7 GENDER feature after solution implementation.

```
0       69
1       74
2       59
3       63
4       63
        ..
1115    57
1116    51
1117    65
1118    57
1119    20
Name: AGE, Length: 1120, dtype: int64
```

```
array(['69', '74', '59', '63', '75', '52', '51', '68', '53', '61', '72',
       '60', '58', '48', '57', '44', '64', '21', '65', '55', '62', '56',
       '67', '77', '70', '54', '49', '73', '47', '71', '66', '76', '78',
       '81', '79', '38', '39', '87', '46', '-56', nan, '170', '35',
       'Twenty One', '43', '28', '20', '22', '32', '27', '40', '37', '25',
       '45', '29', '31', '36', '190', '42', '34', '23', '30', '50', '33',
       '41', '402', '26'], dtype=object)
```

Figure 8 AGE feature before implementation of changes.

Issues resolved:

- Capped outlier ages at 120 and replaced with median.
- Filled missing values with a median.
- Applied absolute function to remove negative values from age.
- Converted strings to numeric.
- Scaled down using min max scaler.

```
0        0.731343
1        0.805970
2        0.582090
3        0.641791
4        0.641791
           ...
1115     0.552239
1116     0.462687
1117     0.671642
1118     0.552239
1119     0.000000
Name: AGE, Length: 1117, dtype: float64
```

```
array([0.73134328, 0.80597015, 0.58208955, 0.64179104, 0.82089552,
       0.47761194, 0.46268657, 0.71641791, 0.49253731, 0.6119403 ,
       0.7761194 , 0.59701493, 0.56716418, 0.41791045, 0.55223881,
       0.35820896, 0.65671642, 0.01492537, 0.67164179, 0.52238806,
       0.62686567, 0.53731343, 0.70149254, 0.85074627, 0.74626866,
       0.50746269, 0.43283582, 0.79104478, 0.40298507, 0.76119403,
       0.68656716, 0.8358209 , 0.86567164, 0.91044776, 0.88059701,
       0.26865672, 0.28358209, 1.        , 0.3880597 , 0.2238806 ,
       0.34328358, 0.11940299, 0.        , 0.02985075, 0.17910448,
       0.10447761, 0.29850746, 0.25373134, 0.07462687, 0.37313433,
       0.13432836, 0.1641791 , 0.23880597, 0.32835821, 0.20895522,
       0.04477612, 0.14925373, 0.44776119, 0.19402985, 0.31343284,
       0.08955224])
```

Figure 9 AGE feature after solution implementation

```
[65] df['DAILY_CIGARETTES']

0        NaN
1         29
2        NaN
3         20
4        NaN
        ...
1115     NaN
1116      21
1117     NaN
1118      21
1119     NaN
Name: DAILY_CIGARETTES, Length: 1120, dtype: object

[66] df['DAILY_CIGARETTES'].isnull().sum()

503
```

```
[67] df['DAILY_CIGARETTES'].unique()

array([nan, '29', '20', '37', '8', '34', '18', '25', '4', '10', '39',
       '33', '14', '31', '36', '2', '32', '21', '13', '23', '40', '38',
       '3', '12', '35', '6', '26', '15', '19', '1', '11', '30', '9',
       '-21', '5', '22', '7', '24', '27', '28', '16', '1000', '17',
       'five'], dtype=object)
```

Figure 10 daily cigarette feature before the implementation of change.

Issues resolved:

- Filled missing values using logical relation (non-smoker = 0 daily cigarettes).
- Converted strings to numeric.
- Capped daily cigarettes at 40 to remove outliers.
- Standardized the scale tomin0 and max 1.

```
newdf['DAILY_CIGARETTES'].isnull().sum()

0

[102] newdf['DAILY_CIGARETTES']

0        0.000
1        0.725
2        0.000
3        0.500
4        0.000
         ...
1115     0.000
1116     0.525
1117     0.000
1118     0.525
1119     0.000
Name: DAILY_CIGARETTES, Length: 1117, dtype: float64
```

```
array([0.   , 0.725, 0.5  , 0.925, 0.2  , 0.85 , 0.45 , 0.625, 0.1  ,
       0.25 , 0.975, 0.825, 0.35 , 0.775, 0.9  , 0.05 , 0.8  , 0.525,
       0.325, 0.575, 1.   , 0.95 , 0.075, 0.3  , 0.875, 0.15 , 0.65 ,
       0.375, 0.475, 0.025, 0.275, 0.75 , 0.225, 0.125, 0.55 , 0.175,
       0.6  , 0.675, 0.7  , 0.4  , 0.425])
```

Figure 11 Daily cigarette feature after solution implementation

```
df['LUNG_CANCER'].unique()

array(['YES', 'NO', 'Yes', 'No', 'nan'], dtype=object)

[104] df['LUNG_CANCER'].isnull().sum()

3
```

Figure 12 Lung cancer feature before implementation of changes.

Issues addressed:

- Standardized values (Yes = 1, No = 0).
- Handled the missing values by dropping them.
- Changed data type to integer.

```
0       1
1       1
2       0
3       0
4       0
       ..
1115    0
1116    0
1117    0
1118    0
1119    0
Name: LUNG_CANCER, Length: 1117, dtype: int64
```

Figure 13 Lung cancer after implementation of changes.

```
COPD_DIAGNOSES          18
FATIGUE                  5
WHEEZING                10
ALCOHOL_CONSUMPTION      0
COUGHING                 0
SHORTNESS_OF_BREATH      4
```

Figure 14 COPD DIAGNOSES ,WHEEZING, FATIGUE, SHORTNESS OF BREATH before implementation of change

Issues addressed:

- Filled missing values with mode.
- Changed data type to integer.

```
[73] df.isnull().sum()

GENDER                  0
AGE                     0
SMOKING_STATUS          0
DAILY_CIGARETTES        0
YELLOW_SKIN             0
ANXIETY                 0
COPD_DIAGNOSES          0
FATIGUE                 0
WHEEZING                0
ALCOHOL_CONSUMPTION     0
COUGHING                0
SHORTNESS_OF_BREATH     0
SWALLOWING_DIFFICULTY   0
CHEST_PAIN              0
LUNG_CANCER             0
```

Figure 15 features after implementation of changes.

YELLOW_SKIN	ANXIETY	COPD_DIAGNOSES	FATIGUE	WHEEZING	ALCOHOL_CONSUMPTION	COUGHING	SHORTNESS_OF_BREATH	SWALLOWING_DIFFICULTY	CHEST_PAIN
2	2	1	2	2	2	2	2	2	2
1	1	2	2	1	1	1	2	2	2
1	1	1	2	2	1	2	2	1	2
2	2	1	1	1	2	1	1	2	2
2	1	1	1	2	1	2	2	1	1

Figure 16 The features before binery enoding.

- **Issues addressed:**

- Labels changed to binary.

YELLOW_SKIN	ANXIETY	COPD_DIAGNOSES	FATIGUE	WHEEZING	ALCOHOL_CONSUMPTION	COUGHING	SHORTNESS_OF_BREATH	SWALLOWING_DIFFICULTY	CHEST_PAIN
1	1	0	1	1	1	1	1	1	1
0	0	1	1	0	0	0	1	1	1
0	0	0	1	1	0	1	1	0	1
1	1	0	0	0	1	0	0	1	1
1	0	0	0	1	0	1	1	0	0

Figure 17 features after binarization.

Task (4) – Modelling: Create Predictive Classification Models

Algorithm Name	Algorithm Type	Learnable Parameters	Some Possible Hyperparameters	Imported Python package to use the algorithm
NB	Parametric	No learnable parameters.	var_smoothing	sklearn.naive_bayes import GaussianNB
ANN (MLP)	Non-parametric	Weights and biases,activation functions	hidden_layer_sizes, activation, solver, alpha, learning_rate, max_iter	from sklearn.neural_network import MLPClassifier
Linear SVM	Parametric	Support vectors, coefficients, intercepts	C, max_iter	from sklearn.svm import SVC
KNN (K=?)	Non-parametric	There is no learnable parameter.	n_neighbors, weights, algorithm, metric, P	from sklearn.neighbors import KNeighborsClassifier

PART B

I, List of retained categorical features.

```
Index(['GENDER', 'SMOKING_STATUS', 'YELLOW_SKIN', 'ANXIETY', 'COPD_DIAGNOSES',
       'FATIGUE ', 'WHEEZING', 'ALCOHOL_CONSUMPTION', 'COUGHING',
       'SHORTNESS_OF_BREATH', 'SWALLOWING_DIFFICULTY', 'CHEST_PAIN'],
      dtype='object')
```

```
Name: LUNG_CANCER, dtype: int64
```

Figure 18 Retained columns

```
print(X_train.shape , y_train.shape)
print(X_test.shape , y_test.shape)
```

```
(781, 12) (781,)
(336, 12) (336,)
```

Figure 19 Data shapes of the retained variables after a test-train split.

II. The 70-30 training-test split ratio is widely recognised in machine learning for its balanced approach to data allocation. According to the book by Kubat (2018), allocating 70% of the data for training enables the model to effectively learn from a substantial sample size, reducing the risk of underfitting. The remaining 30% is reserved for testing, ensuring a robust evaluation of the model's performance and a realistic measure of ability to generalise to new data.

`A 30% test set size will provide us with statistically significant performance metrics by reducing variability and minimizing the impact of random fluctuations within the test data. However, it might face challenges like class imbalance and sample representativeness issues (Liu and Cocea, 2017), which can be mitigated by setting the random state and ensuring stratification in our dataset. This approach, involving training on a significant portion and testing on a separate set, reflects the model's real-world performance with new, unseen data.

Furthermore, empirical evidence and practical applications in machine learning have consistently demonstrated the effectiveness of the 70-30 split across various datasets and models. This ratio is frequently recommended in educational materials and empirical research as a starting point for model development, reflecting its wide acceptance and reliability in the field (Raschka & Mirjalili, 2019).

III. The overall purpose of using a training-test approach is to keep a portion of the data separate from the entire model selection and training process to ensure an unbiased evaluation of the model a simple train-test split is computationally efficient as it involves a single split of the data into training and testing sets, making it quicker to implement. The model is then evaluated using the test dataset to ensure its generalisation ability.

On the other hand, K-fold cross-validation is used to evaluate a model's design rather than a specific training set. It involves repeatedly splitting the dataset into K subsets and then training and testing the model K times on different subsets. This type of evaluation helps to average the model's performance across all the subsets, reducing the risk of overfitting. Cross-validation is particularly useful when there is a limited dataset, as it makes the most of the available data by resampling.

In summary, the training-test split is suitable for initial model evaluation on large datasets, providing a straightforward and quick assessment. In contrast, K-fold cross-validation is preferred for smaller datasets or hyperparameter tuning, offering a thorough evaluation by leveraging multiple data splits. The final test set evaluation ensures the model's performance on truly unseen data, simulating real-world applications and confirming the model's generalisation capability.

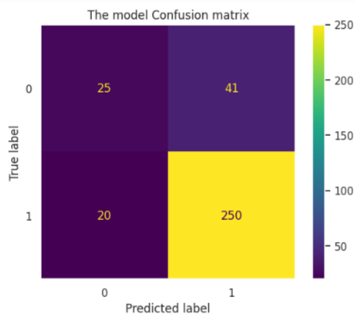
IV.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42, test_size = 0.3, stratify= y )
```

Figure 20 Code snippet to ensure the tested on same test dataset and class distribution ratio is used in our model

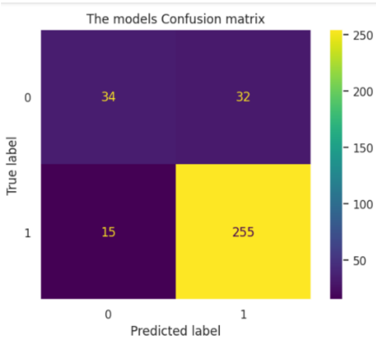
Task (5) – Evaluation:

Part A. Naive Bayes model



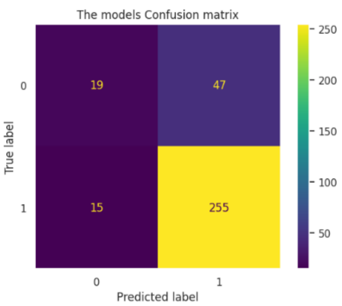

```
[[ 25  41]
 [ 20 250]]
```

KNN classifier model

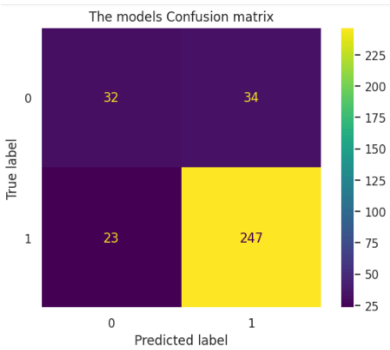



```
[[ 34  32]
 [ 15 255]]
```

Support Vector Machine (SVM) with a Linear Kernel



Artifical neural network MLP




```
[[ 32  34]
 [ 23 247]]
```

Part B.

Metrics	USE or DO	Justification in relation to the success criteria	Model Name	Test Score

	NOT USE			
Accuracy	Don't use	Our aim is not to evaluate the overall performance of the model. Our success criteria are to prioritise the high-risk and low-risk ones. Because our class distribution is imbalanced, accuracy is suitable for balanced distributed classes in the dataset, but it may not be as informative in the case of imbalanced datasets where the number of negatives outweighs the positives.	ANN (MLP)	**83**
			Linear SVM	**82**
			KNN (K=?)	**84**
			NB	**82**
Recall	**Use**	Our emphasis is better identification of high-risk patients(true positives) and the importance of high recall in medical screening, particularly in identifying high-risk patients for undertaking LDCT testing. high recall ensures that the majority of high-risk patients, including those with cancer, are flagged for additional evaluation.	ANN (MLP)	**83**
			Linear SVM	**82**
			KNN (K=?)	**84**
			NB	**82**
Precision	**Use**	Although our top priority is recall, we should also monitor precision to avoid too many false positives, which can lead to unnecessary LCDT tests and other expenses.	ANN (MLP)	**82**
			Linear SVM	**79**
			KNN (K=?)	**83**
			NB	**80**
F-Score	**Don't use**	We don't want the balance between recall and precision. Our desire to minimise false positives and false negatives is not equally important to our research question.	ANN (MLP)	**82**
			Linear SVM	**79**
			KNN (K=?)	**83**
			NB	**80**
AUC-ROC	**Use**	AUC-ROC are useful metrics for evaluating the trade-offs between true and false positive rates. These metrics can help determine how well the model separates the high-risk from low-risk patients.	ANN (MLP)	**70**
			Linear SVM	**61**
			KNN (K=?)	**71**
			NB	**65**

Part C. Suggest the best classification model or models based on the 'USED' performance metrics scores you identified in (Task 5. b). Briefly describe how well your best model satisfies the needs of your healthcare professionals**.**

The KNN model stands out as the best classification model based on its high performance in Recall, precision, and AUC-ROC. With a recall rate of 84%, the model effectively identifies 84% of high-risk patients, reducing the occurrence of false negatives and contributing to early diagnosis and treatment. Additionally, the 83% precision rate indicates that 83% of patients identified as high-risk indeed require further testing, resulting in decreased unnecessary stress and costs.

The AUC-ROC score of 71% demonstrates the model's strong discriminative power, effectively balancing recall and precision to meet clinical needs. Overall, the KNN model's high recall rate supports early detection of lung cancer, while its precise nature optimises resource utilisation and minimises unnecessary procedures.

Part D

i.Indicate the number of cross-validation K folds used

-5 cross-validations k fold used

ii. For the newly tuned model, document the estimated best hyperparameters,

```
knn_gscv.best_params_
```

```
{'algorithm': 'auto', 'n_neighbors': 24, 'p': 1, 'weights': 'distance'}
```

iii Present the test confusion matrix for the best models before and after tuning

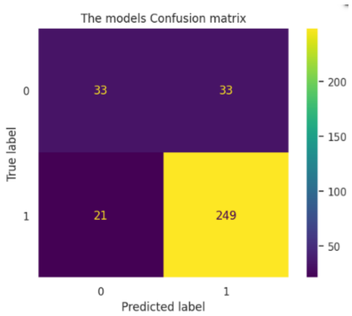


Figure 21Best model's test confusion matrix before tuning

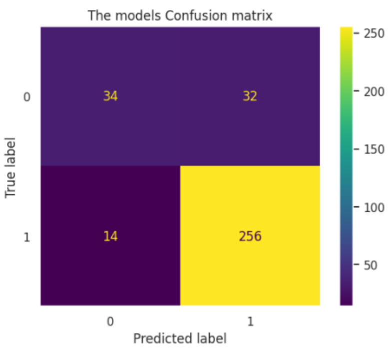
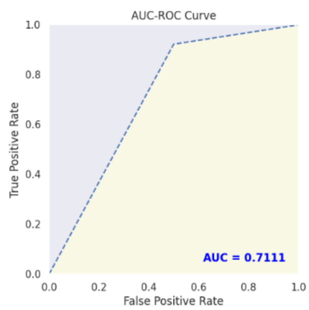


Figure 22 Best model's test confusion matric after tuning

iv. Calculate and document the new score/s of the USED performance metric/s to interpret the success criteria identified in (Task 5.b) before and after tuning.

Before Tuning

	precision	recall	f1-score	support
0	0.61	0.50	0.55	66
1	0.88	0.92	0.90	270
accuracy			0.84	336
macro avg	0.75	0.71	0.73	336
weighted avg	0.83	0.84	0.83	336

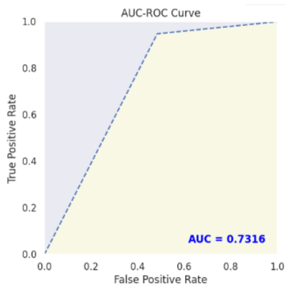


RECALL before tuning: 84

PRECISION before tuning: 83

After Tuning

	precision	recall	f1-score	support
0	0.71	0.52	0.60	66
1	0.89	0.95	0.92	270
accuracy			0.86	336
macro avg	0.80	0.73	0.76	336
weighted avg	0.85	0.86	0.85	336



RECALL after tuning: 86

PRECISION after tuning: 85

v. Explain your observations on whether the tuning of hyperparameters enhanced the generalisation of your original best model in line with the success criteria.

-Tuning of the hyperparameters improved our performance metrics values recall from 84% to 86%, the precision of 83% to 85%, and AUC-ROC of 71% to 73%. This means our tuned model has effectively identified high-risk patients, and our model's ability to generalise has improved, meaning it performs better on unseen data and accurately identifies more true positives, additionally reduces false positives, minimising unnecessary stress and additional expenses for patients and making it more effective and reliable for lung cancer screening.

Part E, Based on your best model, draft an answer for the research question, criticise your best-performing model, and state any limitations you may have identified. Research and try to explain why your selected algorithm overtook all other models in no more than 200 words. State any ethical issues your model may raise if used to screen lung cancer.

-Based on the K-Nearest Neighbors (KNN) model, machine learning shows promise in creating a non-invasive, inexpensive screening tool for predicting those who need LDCT testing for lung cancer. Our best model achieved an impressive recall of 86%, precision of 85%, and an AUC-ROC of 73%, indicating strong performance in identifying high-risk patients. However, the model is not without limitations. The KNN algorithm, while effective, is computationally intensive and sensitive to the choice of K and distance metrics, which can impact performance on larger datasets. The model's reliance on existing data patterns means it may struggle with unseen variations. KNN outperformed other models due to its simplicity and effectiveness in handling medium-sized datasets. Ethical concerns include concerns about data privacy and security, emphasising the need for proper handling of sensitive medical information. Additionally, false positives, though reduced, still pose a risk of unnecessary expense of 1310 £ per false positives, stress and medical procedures for patients. Addressing these limitations and ethical issues is crucial for deploying a reliable and fair screening tool.

Reference list

American cancer society (2019). *Changes in Skin Color | Skin Problems*. [online] www.cancer.org. Available at: https://www.cancer.org/cancer/managing-cancer/side-effects/hair-skin-nails/skin-color-changes.html.

Carnio, S., Di Stefano, R. and Novello, S. (2016). Fatigue in lung cancer patients: symptom burden and management of challenges. *Lung Cancer: Targets and Therapy*, 7, p.73. doi:https://doi.org/10.2147/lctt.s85334.

Eldridge, L. (2023). *How Lung Cancer Affects Different Age Groups*. [online] Verywell Health. Available at: https://www.verywellhealth.com/lung-cancer-age-5216079#:~:text=Like%20most%20cancer%20types%2C%20the%20chance%20of%20developing.

Houghton, A.M. (2013). Mechanistic links between COPD and lung cancer. *Nature Reviews Cancer*, 13(4), pp.233–245. doi:https://doi.org/10.1038/nrc3477.

Jafri, S.H.R., Ali, F., Mollaeian, A., Hasan, S.M., Hussain, R., Akkanti, B.H., Williams, J.T., Advani, S.M. and El-Osta, H.E. (2017). Major stressful life events and risk of developing lung cancer. *Journal of Clinical Oncology*, 35(15_suppl), pp.1575–1575. doi:https://doi.org/10.1200/jco.2017.35.15_suppl.1575.

John Hopkins Medicine (2019). *Manage Shortness of Breath with Lung Cancer*. [online] John Hopkins Medicine. Available at: https://www.hopkinsmedicine.org/health/conditions-and-diseases/lung-cancer/manage-shortness-of-breath-with-lung-cancer.

Kubat, M. (2018). *Introduction To Machine Learning.* S.L.: Springer International Pu.

Liu, H. and Cocea, M. (2017). Semi-random partitioning of data into training and test sets in granular computing context. *Granular Computing*, 2(4), pp.357–386. doi:https://doi.org/10.1007/s41066-017-0049-2.

May, L., Shows, K., Nana-Sinkam, P., Li, H. and Landry, J.W. (2023). Sex Differences in Lung Cancer. *Cancers*, [online] 15(12), pp.3111–3111. doi:https://doi.org/10.3390/cancers15123111.

National Health Service (2022). *Overview - Lung cancer*. [online] NHS. Available at: https://www.nhs.uk/conditions/lung-cancer/.

Tam, A. (2021). *Training-validation-test split and cross-validation done right*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/training-validation-test-split-and-cross-validation-done-right/.

Thompson, A., Cook, J., Choquet, H., Jorgenson, E., Yin, J., Kinnunen, T., Barclay, J., Morris, A.P. and Pirmohamed, M. (2020). Functional validity, role, and implications of heavy alcohol consumption genetic loci. *Science Advances*, [online] 6(3), p.eaay5034. doi:https://doi.org/10.1126/sciadv.aay5034.

www.minitab.com. (n.d.). *Data Mining, Machine Learning & Predictive Analytics Software | Minitab*. [online] Available at: http://info.salford-systems.com/blog/bid/337783/Why-Data-Scientists-Split-Data-into-Train-and-Test [Accessed 1 Jul. 2024].

Yang, H., Tan, Z., Zhang, Y., Sun, J. and Huang, P. (2022). ABO blood classification and the risk of lung cancer: A meta-analysis and trial sequential analysis. *Oncology Letters*, 24(4). doi:https://doi.org/10.3892/ol.2022.13460.