Random Forest Model for Lung Cancer

March 10, 2025

[86]:
```
 ## Load required libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import accuracy_score, classification_report,␣
 ↪confusion_matrix
from sklearn.ensemble import RandomForestClassifier
import joblib
from sklearn.preprocessing import LabelEncoder
import warnings
warnings.filterwarnings("ignore")
```

[87]:
```
## Import the dataset
df = pd.read_csv("C:/Users/PC/OneDrive/Desktop/Data Science/Datasets/Datasets/
 ↪survey lung cancer_045236.csv")
```

[88]:
```
## View the first 10 observations of the dataset
df.head(10)
```

[88]:
```
  GENDER  AGE  SMOKING  YELLOW_FINGERS  ANXIETY  PEER_PRESSURE  \
0      M   69        1               2        2              1
1      M   74        2               1        1              1
2      F   59        1               1        1              2
3      M   63        2               2        2              1
4      F   63        1               2        1              1
5      F   75        1               2        1              1
6      M   52        2               1        1              1
7      F   51        2               2        2              2
8      F   68        2               1        2              1
9      M   53        2               2        2              2

   CHRONIC DISEASE  FATIGUE   ALLERGY   WHEEZING  ALCOHOL CONSUMING  COUGHING  \
0                1         2         1         2                  2         2
1                2         2         2         1                  1         1
2                1         2         1         2                  1         2
3                1         1         1         1                  2         1
```

```
4                1         1         1         2                  1         2
5                2         2         2         2                  1         2
6                1         2         1         2                  2         2
7                1         2         2         1                  1         1
8                1         2         1         1                  1         1
9                2         1         2         1                  2         1

   SHORTNESS OF BREATH  SWALLOWING DIFFICULTY  CHEST PAIN LUNG_CANCER
0                    2                      2           2         YES
1                    2                      2           2         YES
2                    2                      1           2          NO
3                    1                      2           2          NO
4                    2                      1           1          NO
5                    2                      1           1         YES
6                    2                      1           2         YES
7                    2                      2           1         YES
8                    1                      1           1          NO
9                    1                      2           2         YES
```

[89]:

```
## Inspect the structure of the dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 309 entries, 0 to 308
Data columns (total 16 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   GENDER                 309 non-null    object
 1   AGE                    309 non-null    int64
 2   SMOKING                309 non-null    int64
 3   YELLOW_FINGERS         309 non-null    int64
 4   ANXIETY                309 non-null    int64
 5   PEER_PRESSURE          309 non-null    int64
 6   CHRONIC DISEASE        309 non-null    int64
 7   FATIGUE                309 non-null    int64
 8   ALLERGY                309 non-null    int64
 9   WHEEZING               309 non-null    int64
 10  ALCOHOL CONSUMING      309 non-null    int64
 11  COUGHING               309 non-null    int64
 12  SHORTNESS OF BREATH    309 non-null    int64
 13  SWALLOWING DIFFICULTY  309 non-null    int64
 14  CHEST PAIN             309 non-null    int64
 15  LUNG_CANCER            309 non-null    object
dtypes: int64(14), object(2)
memory usage: 38.8+ KB
```

[90]:

```
## Check the data types
df.dtypes
```

[90]:

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

[91]:

```
## Check the unique entries of the outcome variable
print(df["LUNG_CANCER"].unique())
```

```
['YES' 'NO']
```

[92]:

```
## Check target class distribution
print(df["LUNG_CANCER"].value_counts())
```

```
LUNG_CANCER
YES    270
NO      39
Name: count, dtype: int64
```

[93]:

```
## Check for duplicates
df.duplicated().sum()
```

[93]:

```
33
```

[94]:

```
## Check for missing values
df.isnull().sum()
```

[94]:

```
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
```

[95]:

```
## Data Preprocessing
df = df.drop_duplicates()
```

[96]:

```
## Label encoding
## Select categorical columns
categorical_cols = df.select_dtypes(include = ["object"]).columns
## Initialize the label encoder
label_encoder = LabelEncoder()
## Apply label encooding to selected columns
for col in categorical_cols:
    df[col] = label_encoder.fit_transform(df[col])
```

[98]:

```
## Define the target and the features
X = df.drop(columns = ["LUNG_CANCER"])
y = df["LUNG_CANCER"]
```

[99]:

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

## Check the shape of the training and testing sets
print(X_train.shape, X_test.shape)
```

```
(220, 15) (56, 15)
```

[100]:

```
## Train the Random Forest Model
## Initialize Random Forest Classifier
rf_classifier = RandomForestClassifier(n_estimators = 100, random_state = 42)

## Train the model
rf_classifier.fit(X_train, y_train)
```

[100]:

```
RandomForestClassifier(random_state=42)
```

[101]:

```
## Make predictions
y_pred = rf_classifier.predict(X_test)
```

[102]:

```
## Model Evaluation
## Model Accuracy
```

```
print("Accuracy:", accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.9285714285714286
```

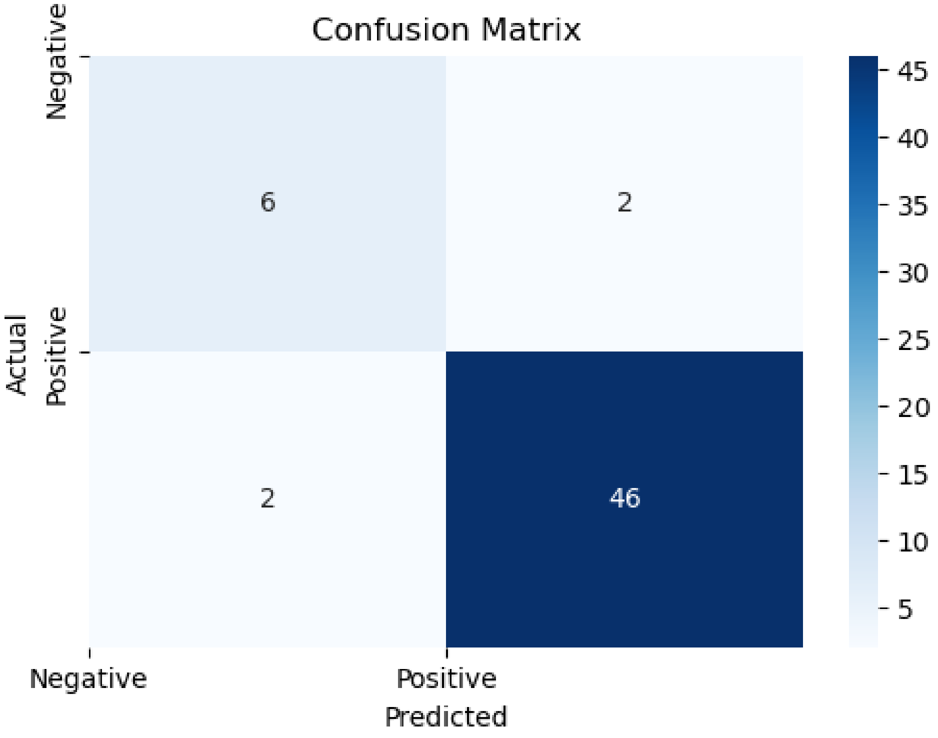
[103]:

```
## Classification Report
print(classification_report(y_test, y_pred))
```

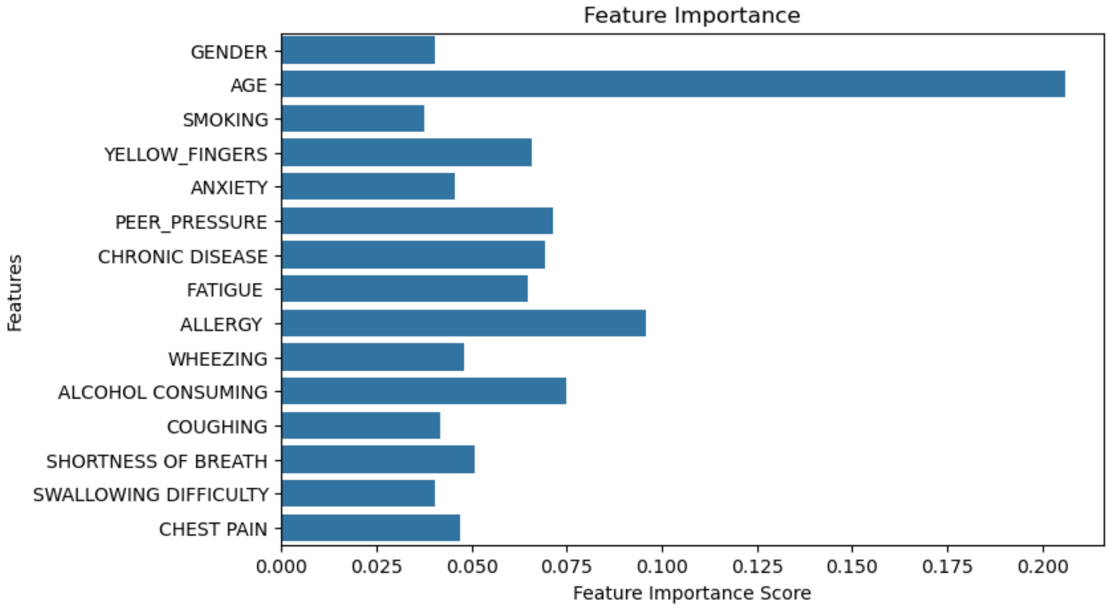
```
              precision    recall  f1-score   support

           0       0.75      0.75      0.75         8
           1       0.96      0.96      0.96        48

    accuracy                           0.93        56
   macro avg       0.85      0.85      0.85        56
weighted avg       0.93      0.93      0.93        56
```

[104]:

```
## Confusion Matrix
## Create a confusion matrix
cm = confusion_matrix(y_test, y_pred)

## Visualizing Confusion Matrix
plt.figure(figsize = (6, 4))
sns.heatmap(cm, annot = True, fmt = "d", cmap = "Blues")
plt.xlabel("Predicted")
plt.xticks([0, 1], labels = ["Negative", "Positive"])
plt.yticks([0, 1], labels = ["Negative", "Positive"])
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()
```



[105]:
```
## Feature importance
## Extract Feature importance
feature_importance = rf_classifier.feature_importances_
feature_names = X.columns

## Plot feature importance
plt.figure(figsize = (8, 5))
sns.barplot( x = feature_importance, y = feature_names)
plt.xlabel("Feature Importance Score")
plt.ylabel("Features")
plt.title("Feature Importance")
plt.show()
```



[106]:

```python
## Save the model
Lung_cancer_model = rf_classifier
joblib.dump(Lung_cancer_model, "rf_classifier.pkl")
```

[106]:

```
['rf_classifier.pkl']
```

[107]:

```python
## Load the model
joblib.load("rf_classifier.pkl")
```

[107]:

```
RandomForestClassifier(random_state=42)
```