Stephen Marth

ISE-221

Final Project Task 1

For my ISE-221 final project, I have chosen a publicly accessible dataset from Kaggle. This dataset contains 16 features across 5,000 samples, including a target variable. The goal is to identify features that may be indicative of pulmonary disease. Since the target variable is binary (yes or no), I have decided that the best approach is to develop a binary classification model that will predict the likelihood of pulmonary disease based on the selected features.

Shortly after starting my build, I realized that I wanted this program to be highly adaptable to other comma-separated values (CSV) files. The main requirements for compatibility are:

1. The target variable must be in the last column.
2. The first row must contain the feature names.
3. Each column represents a feature, and each row represents a sample.

In Task 1, the following has been accomplished:

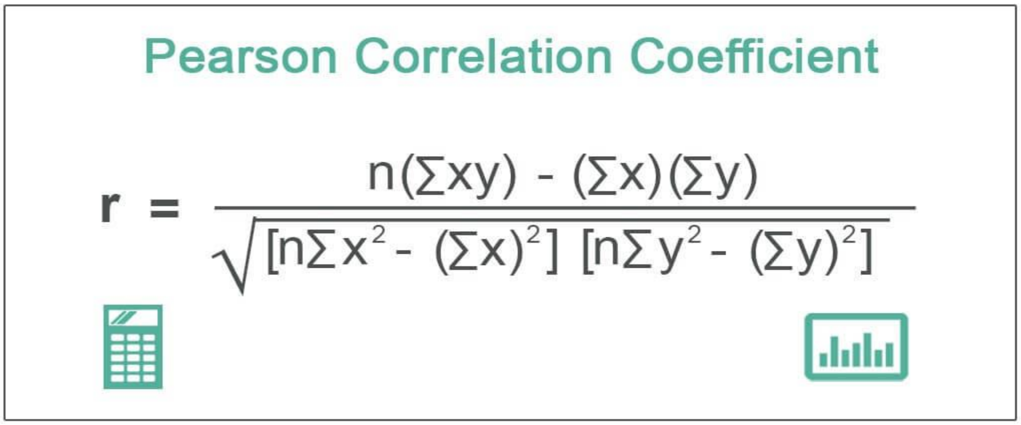
Download the Dataset and clean it

I created the variables, file_path and file_name to allow easy modification of the file path when needed. Then, I used np.genfromtxt() to load the CSV file. However, since genfromtxt() automatically replaces non-numeric values with NaN, I manually replaced all "YES" and "NO" values with their binary equivalents: 1 and 0. After this conversion, I changed the dataset to a float type to ensure compatibility with future numerical computations.

Once all transformations were completed, a report was generated displaying the number of features, the number of samples, and a list of feature names alongside their index numbers for easy reference.

Pick the features for use as inputs

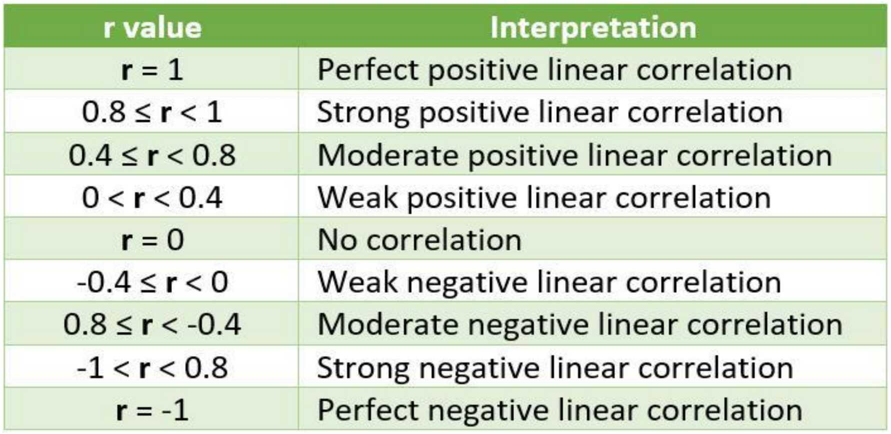
The dataset originally contained 17 features along with a target variable. To determine which features to use as inputs for the model, I applied the Pearson correlation coefficient. The Pearson correlation coefficient or "r" value measures the linear relationship between each feature and the target variable, Pulmonary Disease. Features with higher absolute correlation values were prioritized, as they have a stronger relationship with the target and are more likely to improve model performance.



Pearson Correlation Coefficient

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[n\Sigma x^2 - (\Sigma x)^2]\,[n\Sigma y^2 - (\Sigma y)^2]}}$$

According to the correlation strength reference chart, no feature exhibited a strong correlation (|r| > 0.8). However, several features had moderate or weak correlations, which may still contribute useful predictive power. Based on this, I set a lower threshold (|r| > 0.15) to retain features that showed at least some relationship with the target variable.



r value	Interpretation
r = 1	Perfect positive linear correlation
0.8 ≤ **r** < 1	Strong positive linear correlation
0.4 ≤ **r** < 0.8	Moderate positive linear correlation
0 < **r** < 0.4	Weak positive linear correlation
r = 0	No correlation
-0.4 ≤ **r** < 0	Weak negative linear correlation
0.8 ≤ **r** < -0.4	Moderate negative linear correlation
-1 < **r** < 0.8	Strong negative linear correlation
r = -1	Perfect negative linear correlation

The top six features with a correlation above .15

FEATURE	“r” Value
SMOKING	0.4615
SMOKING_FAMILY_HISTORY	0.3025
THROAT_DISCOMFORT	0.2835
BREATHING_ISSUE	0.2705
STRESS_IMMUNE	0.1811
ENERGY_LEVEL	0.1715

Task 1 outputs figure 1-3

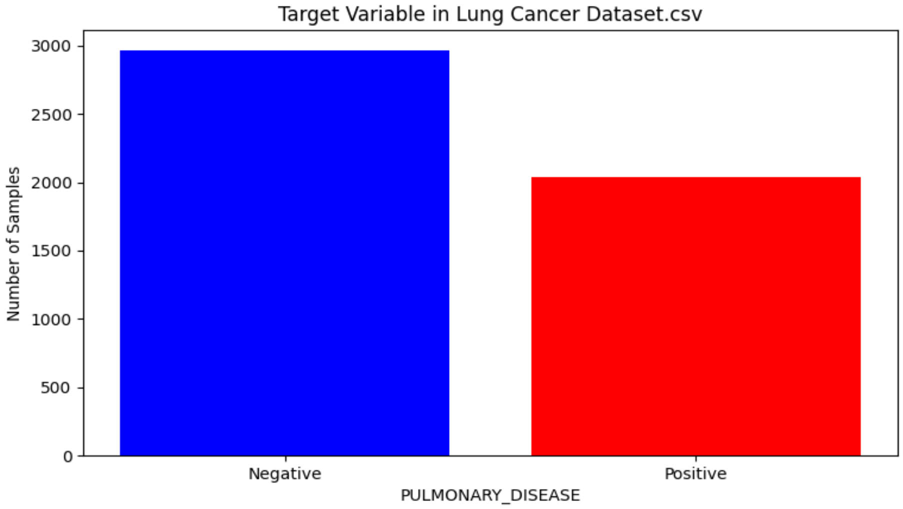


Figure 1

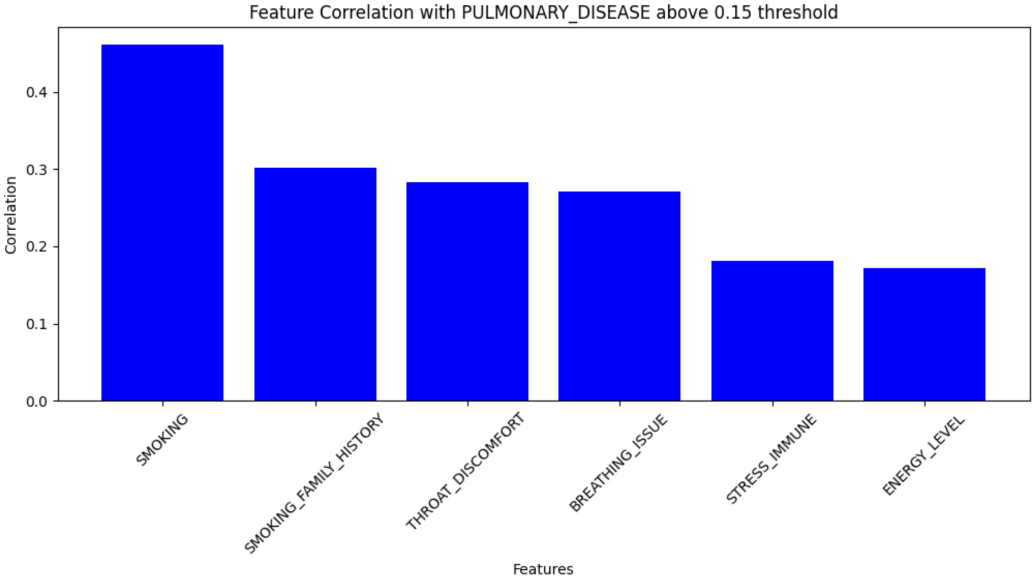
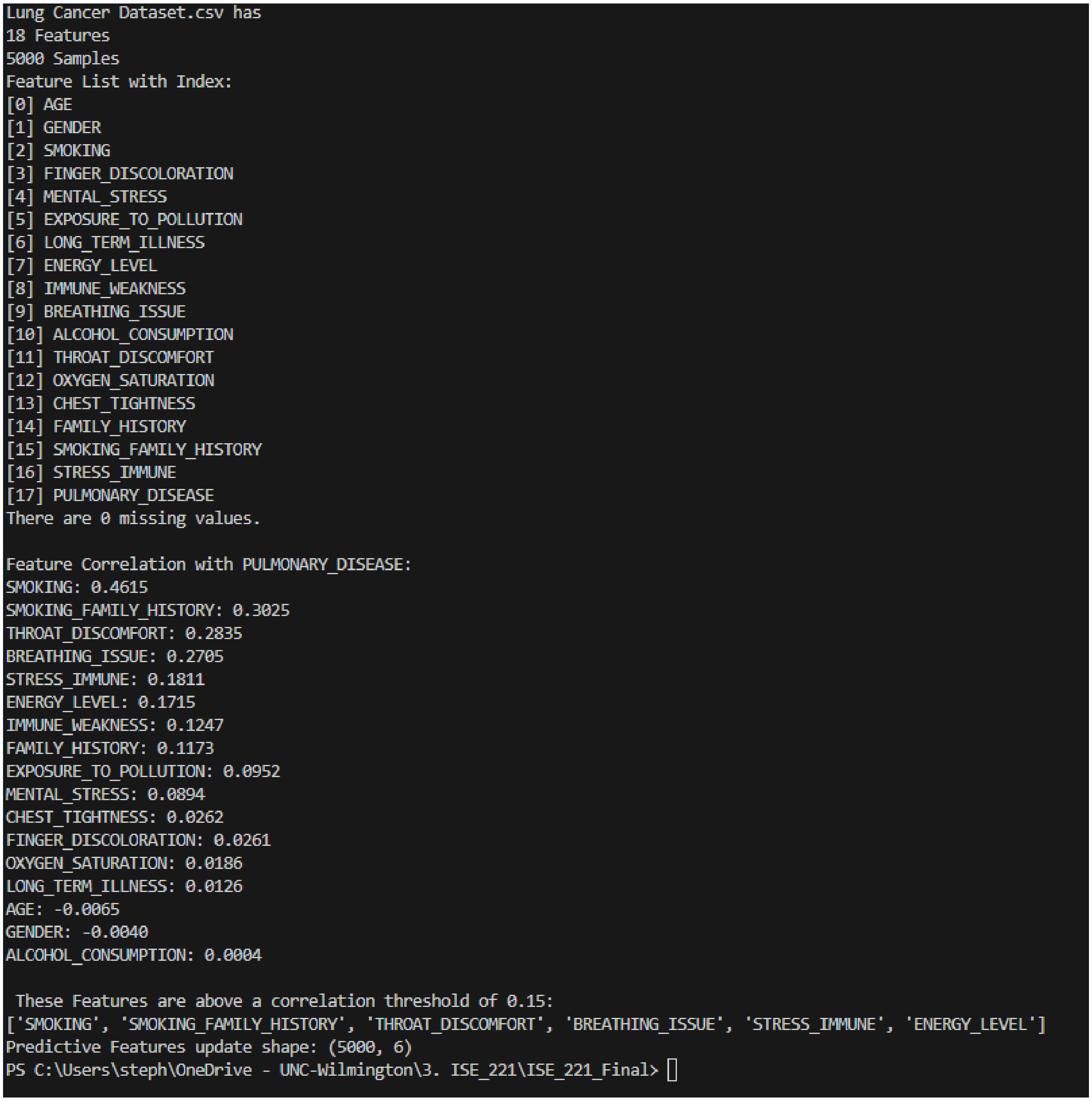


Figure 2

Output



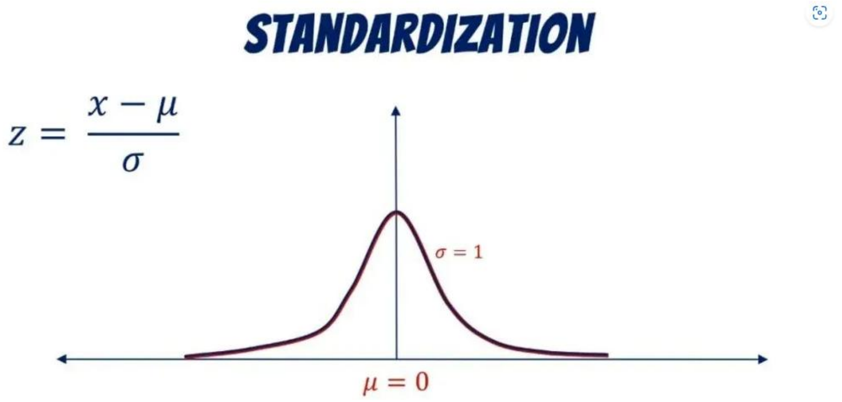
```
Lung Cancer Dataset.csv has
18 Features
5000 Samples
Feature List with Index:
[0] AGE
[1] GENDER
[2] SMOKING
[3] FINGER_DISCOLORATION
[4] MENTAL_STRESS
[5] EXPOSURE_TO_POLLUTION
[6] LONG_TERM_ILLNESS
[7] ENERGY_LEVEL
[8] IMMUNE_WEAKNESS
[9] BREATHING_ISSUE
[10] ALCOHOL_CONSUMPTION
[11] THROAT_DISCOMFORT
[12] OXYGEN_SATURATION
[13] CHEST_TIGHTNESS
[14] FAMILY_HISTORY
[15] SMOKING_FAMILY_HISTORY
[16] STRESS_IMMUNE
[17] PULMONARY_DISEASE
There are 0 missing values.

Feature Correlation with PULMONARY_DISEASE:
SMOKING: 0.4615
SMOKING_FAMILY_HISTORY: 0.3025
THROAT_DISCOMFORT: 0.2835
BREATHING_ISSUE: 0.2705
STRESS_IMMUNE: 0.1811
ENERGY_LEVEL: 0.1715
IMMUNE_WEAKNESS: 0.1247
FAMILY_HISTORY: 0.1173
EXPOSURE_TO_POLLUTION: 0.0952
MENTAL_STRESS: 0.0894
CHEST_TIGHTNESS: 0.0262
FINGER_DISCOLORATION: 0.0261
OXYGEN_SATURATION: 0.0186
LONG_TERM_ILLNESS: 0.0126
AGE: -0.0065
GENDER: -0.0040
ALCOHOL_CONSUMPTION: 0.0004

 These Features are above a correlation threshold of 0.15:
['SMOKING', 'SMOKING_FAMILY_HISTORY', 'THROAT_DISCOMFORT', 'BREATHING_ISSUE', 'STRESS_IMMUNE', 'ENERGY_LEVEL']
Predictive Features update shape: (5000, 6)
PS C:\Users\steph\OneDrive - UNC-Wilmington\3. ISE_221\ISE_221_Final> 
```

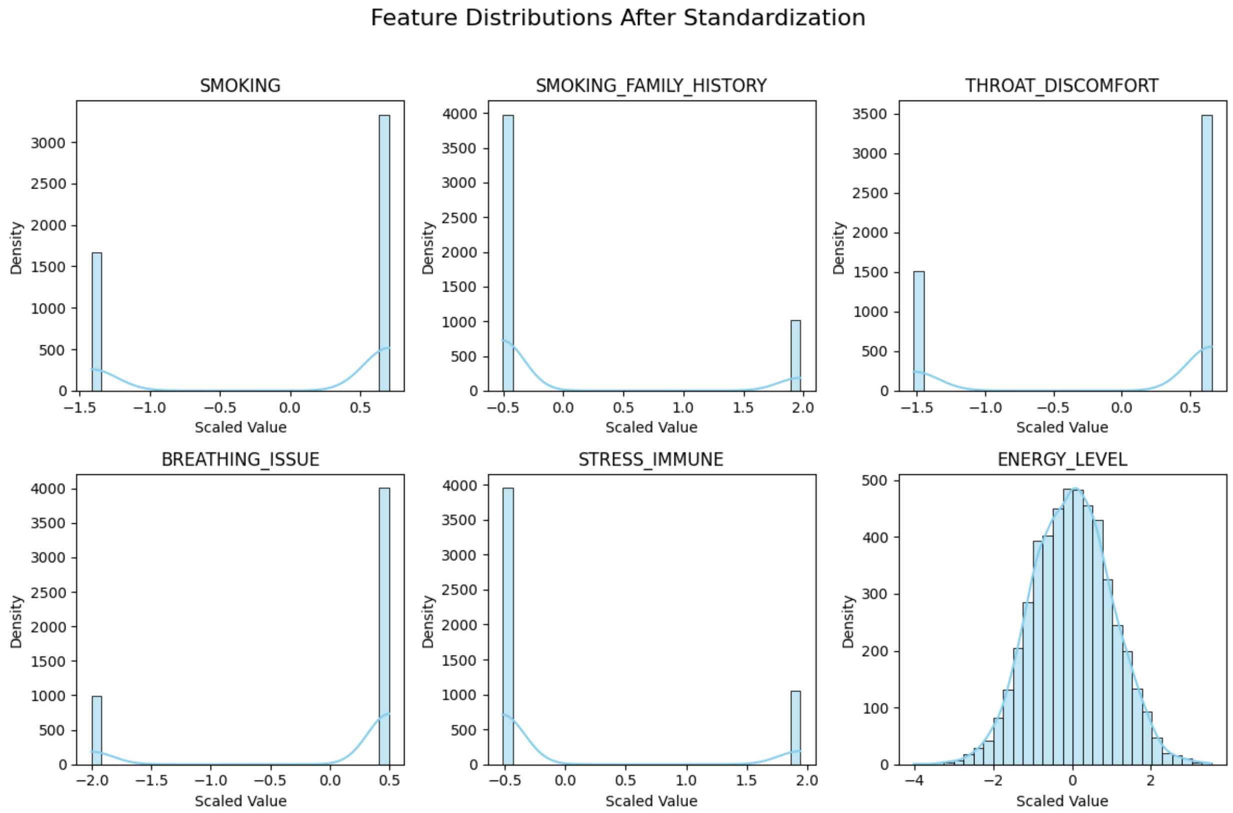
Figure 3

TASK 2

Scaling the Data

Standardization was selected because each feature in the dataset is measured on a different scale. Standardization (also known as Z-score scaling) transforms the features so that they are centered around zero and have a standard deviation of one. This ensures that all features contribute equally during model training, preventing those with larger numeric ranges from dominating the learning process.





After standardization, the feature distributions show that most features) remain concentrated at the far ends of unit variance due to their binary nature. However, ENERGY_LEVEL demonstrates a near-normal distribution, confirming successful scaling of the data set.

The dataset was split into three separate categories:

1. **Training Data** – This set needs to be large enough so the model can learn and build general parameters. It should capture the underlying patterns in the data. If this set is too small, it can pose a risk of underfitting — meaning the model won't learn enough to make accurate predictions.

2. **Validation Data** – This set is used to fine-tune hyperparameters, evaluate different model choices, and help decide when to stop training. It plays a key role in preventing overfitting by monitoring model performance during training.

3. **Testing Data** – This set provides completely unseen data for the model to evaluate its final performance. It allows us to verify how well the model generalizes to new, real-world inputs.

A 60% training, 20% validation, 20% testing split, a well-established best practice in machine learning is used. It ensures that the model has sufficient data to learn from, an independent set to tuning, and a clean test set for final performance evaluation.

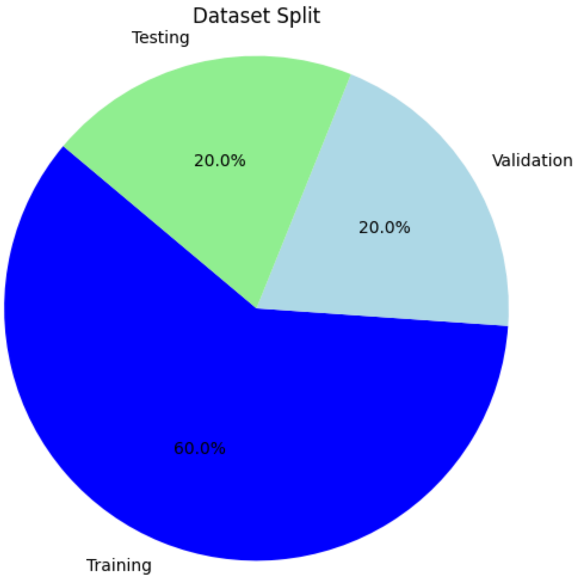


Figure 4