Lung Cancer Analysis Project Report

1. Introduction

Lung cancer is a significant health challenge worldwide. Early detection and accurate prediction of lung cancer can save lives by allowing timely interventions. In this project, we aim to predict lung cancer using various machine-learning techniques based on clinical data. The process involves data cleaning, exploratory data analysis (EDA), feature engineering, and model evaluation.

2. Data Preprocessing

What is Data Preprocessing?

Data preprocessing is a critical step in any machine learning project. It involves cleaning the dataset, handling missing values, encoding categorical variables, and normalizing or scaling the features.

Preprocessing Steps in the Project:

1. **Handling Missing Values:**
   - The dataset was checked for missing values using methods like .isnull().sum(). If any missing values were present, strategies like imputation or removal were applied.
2. **Encoding Categorical Variables:**
   - Categorical data were transformed into a machine-readable format using techniques such as One-Hot Encoding or Label Encoding to convert the non-numeric columns into numeric values.
3. **Normalization:**
   - Feature scaling was performed using the StandardScaler() to ensure that the features are on a similar scale. This step is essential for distance-based algorithms like K-Nearest Neighbors (KNN).

3. Exploratory Data Analysis (EDA)

What is EDA?

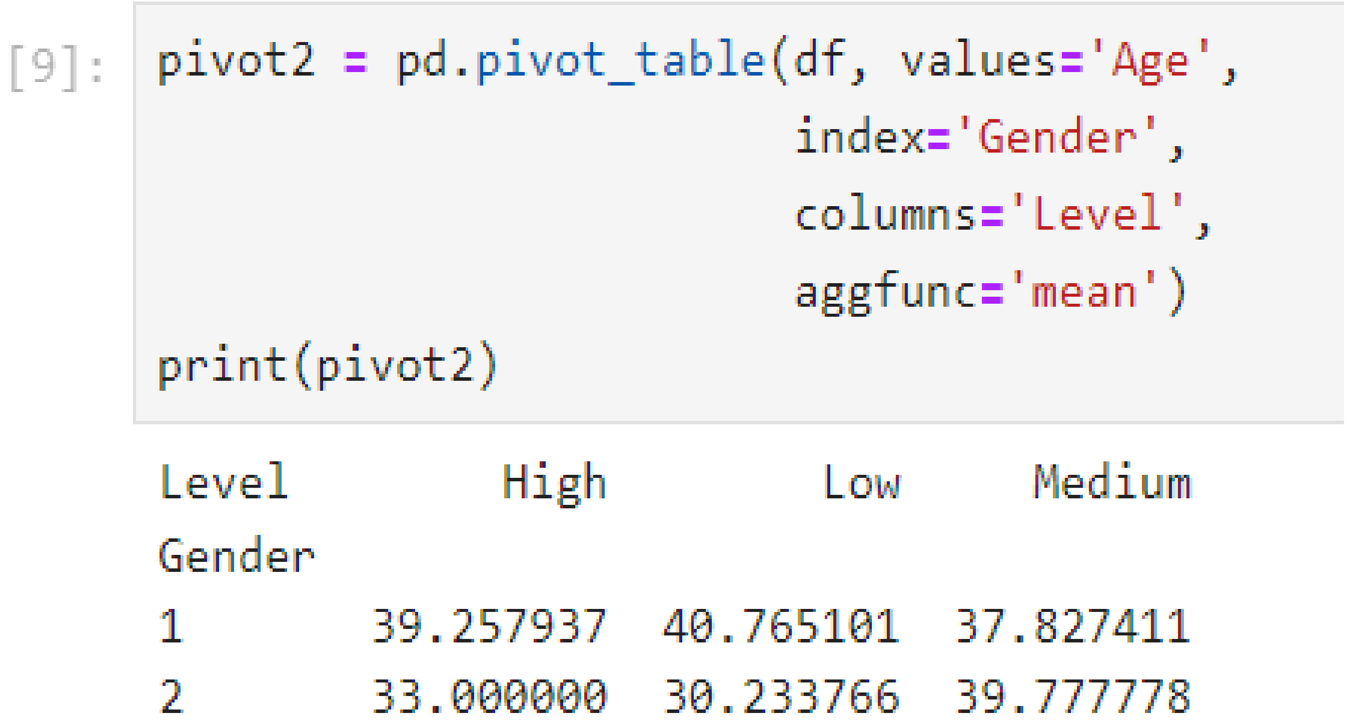
EDA is a technique used to analyze datasets and summarize their main characteristics, often using visual methods. It helps in identifying patterns, relationships, outliers, and important features that can influence model performance.

EDA in This Project:

1. **Distribution of Features:**
   - The data's distribution was visualized using histograms and box plots to identify skewness or outliers.
   - Important features like age, smoking habits, and tumor size were examined for their influence on lung cancer occurrence.

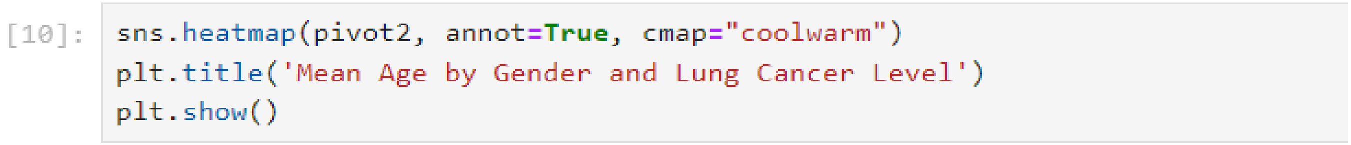
Example:

Pivot Table & Chart

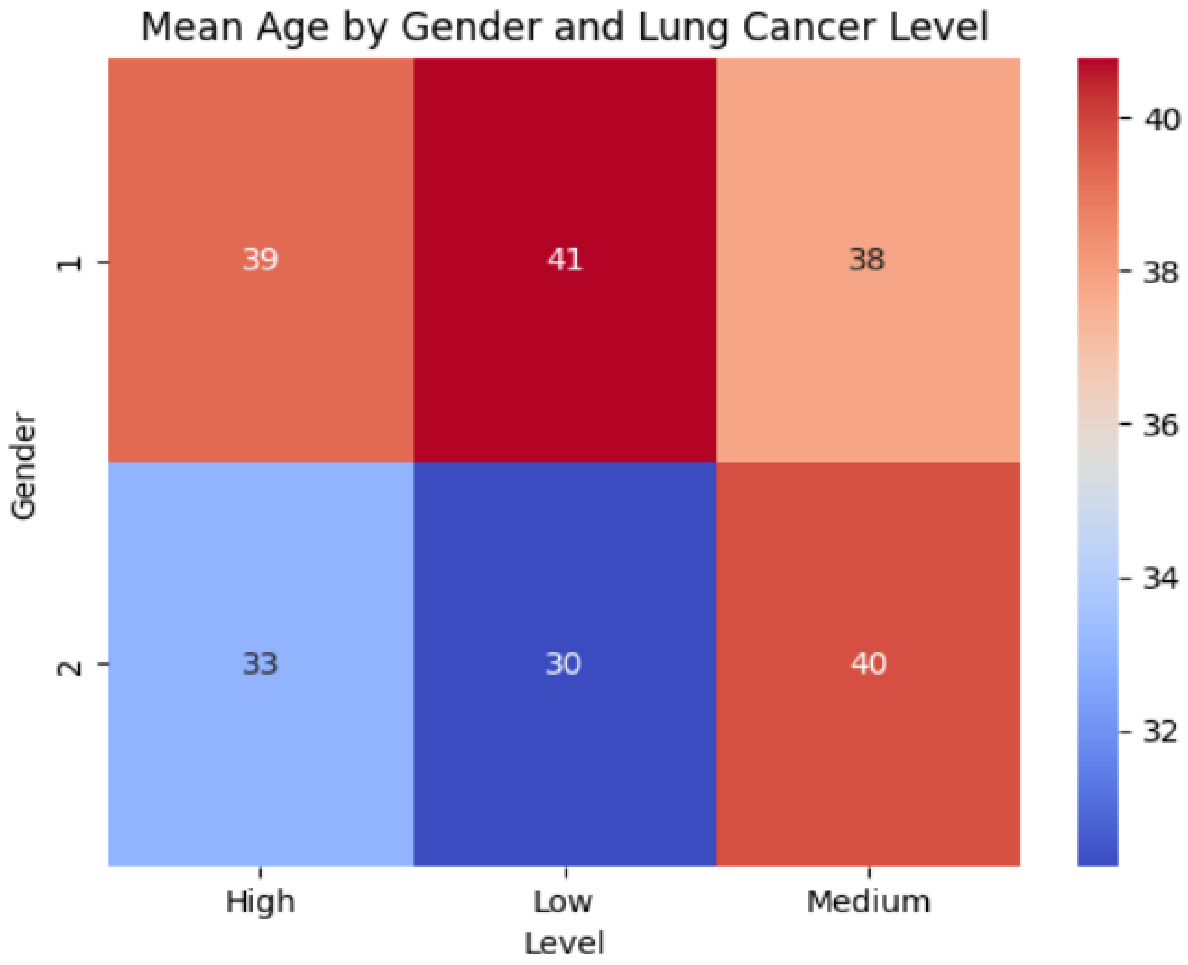


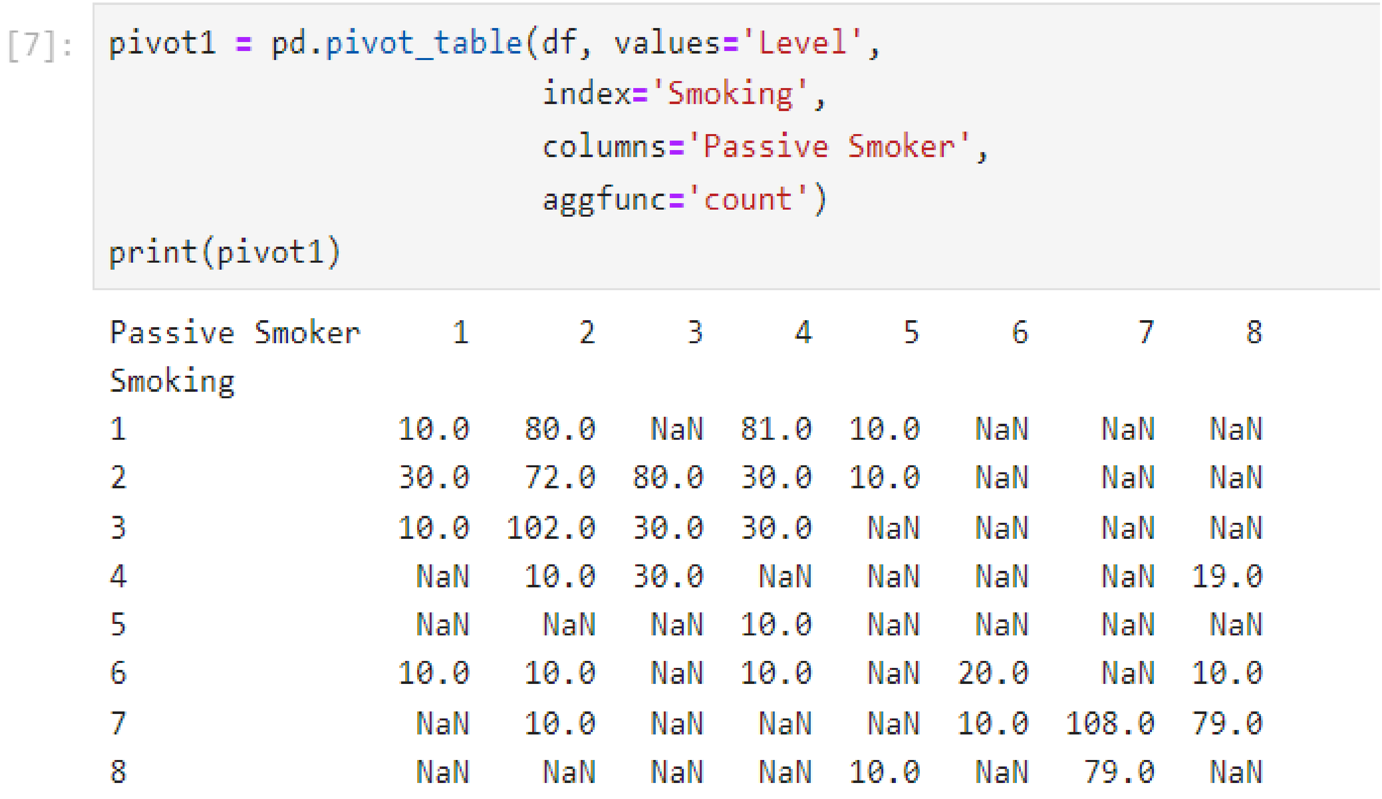
```
[9]: pivot2 = pd.pivot_table(df, values='Age',
                              index='Gender',
                              columns='Level',
                              aggfunc='mean')
     print(pivot2)
```

```
Level        High        Low     Medium
Gender
1       39.257937  40.765101  37.827411
2       33.000000  30.233766  39.777778
```



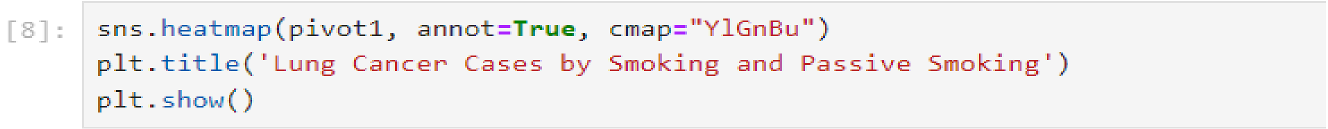
```
[10]: sns.heatmap(pivot2, annot=True, cmap="coolwarm")
      plt.title('Mean Age by Gender and Lung Cancer Level')
      plt.show()
```



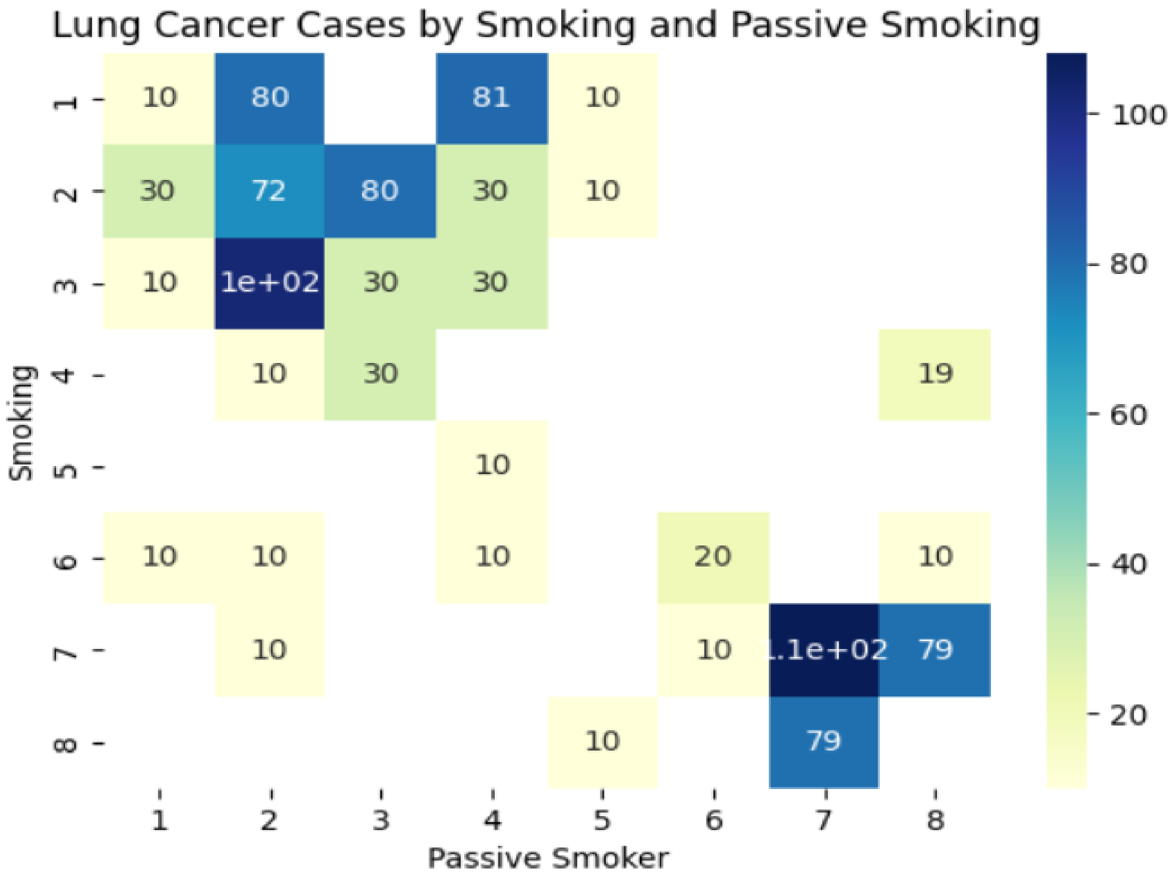


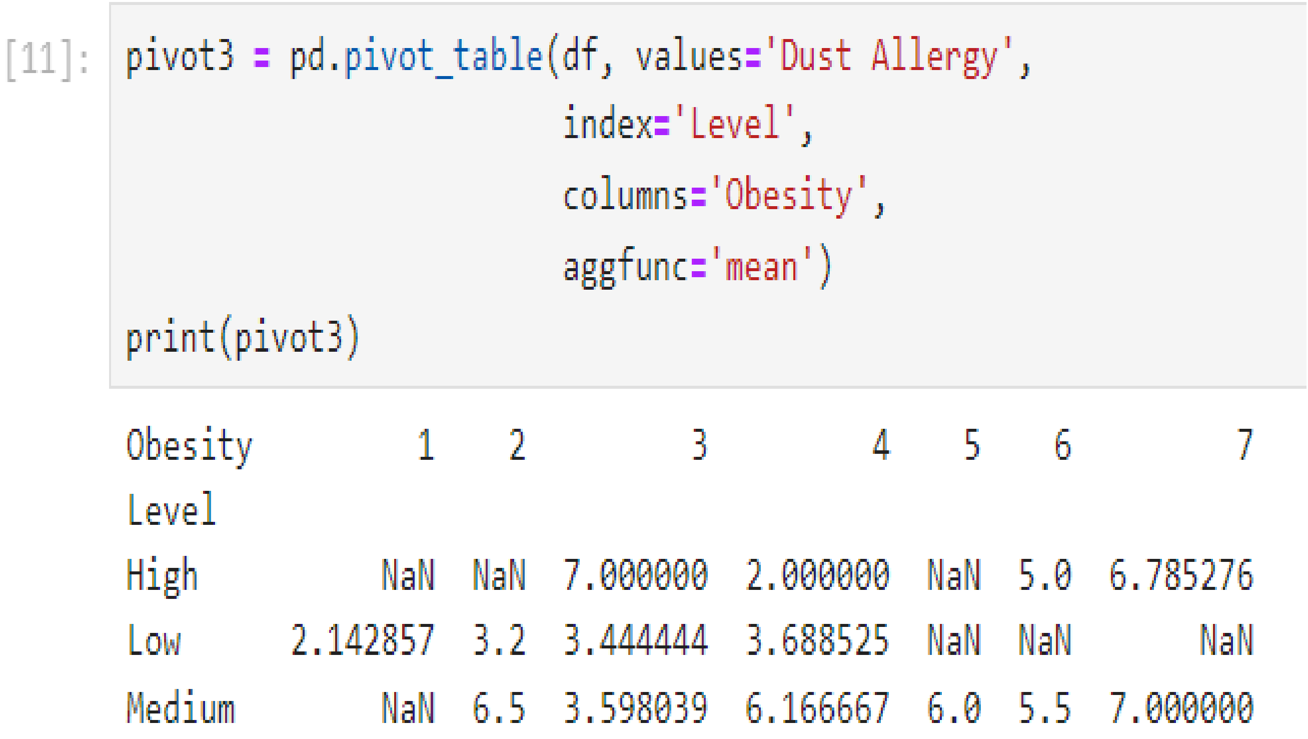
```
[7]: pivot1 = pd.pivot_table(df, values='Level',
                            index='Smoking',
                            columns='Passive Smoker',
                            aggfunc='count')
     print(pivot1)
```

```
Passive Smoker     1      2     3     4     5     6      7     8
Smoking
1               10.0   80.0   NaN  81.0  10.0   NaN    NaN   NaN
2               30.0   72.0  80.0  30.0  10.0   NaN    NaN   NaN
3               10.0  102.0  30.0  30.0   NaN   NaN    NaN   NaN
4                NaN   10.0  30.0   NaN   NaN   NaN    NaN  19.0
5                NaN    NaN   NaN  10.0   NaN   NaN    NaN   NaN
6               10.0   10.0   NaN  10.0   NaN  20.0    NaN  10.0
7                NaN   10.0   NaN   NaN   NaN  10.0  108.0  79.0
8                NaN    NaN   NaN   NaN  10.0   NaN   79.0   NaN
```



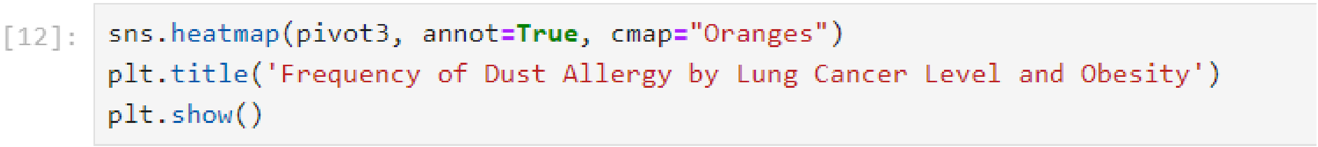
```
[8]: sns.heatmap(pivot1, annot=True, cmap="YlGnBu")
     plt.title('Lung Cancer Cases by Smoking and Passive Smoking')
     plt.show()
```



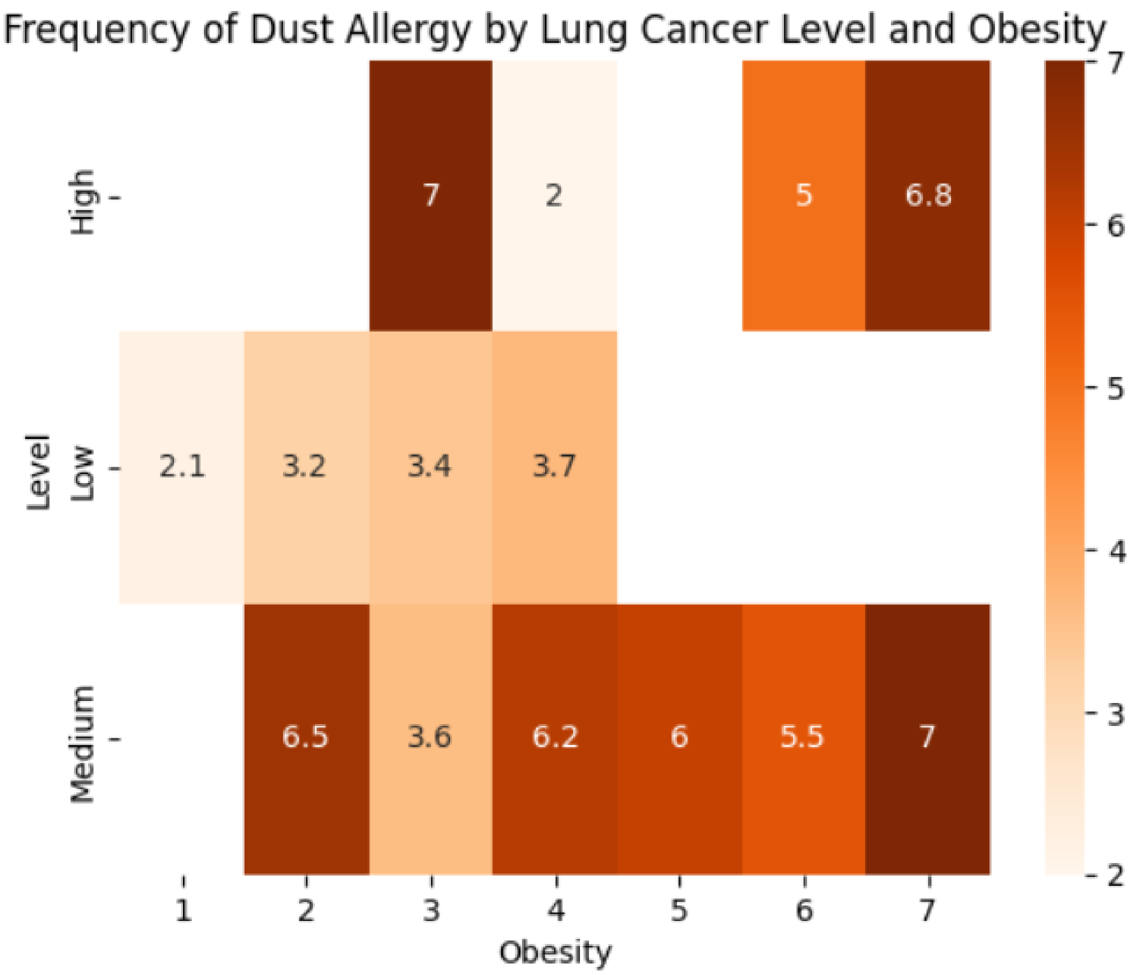


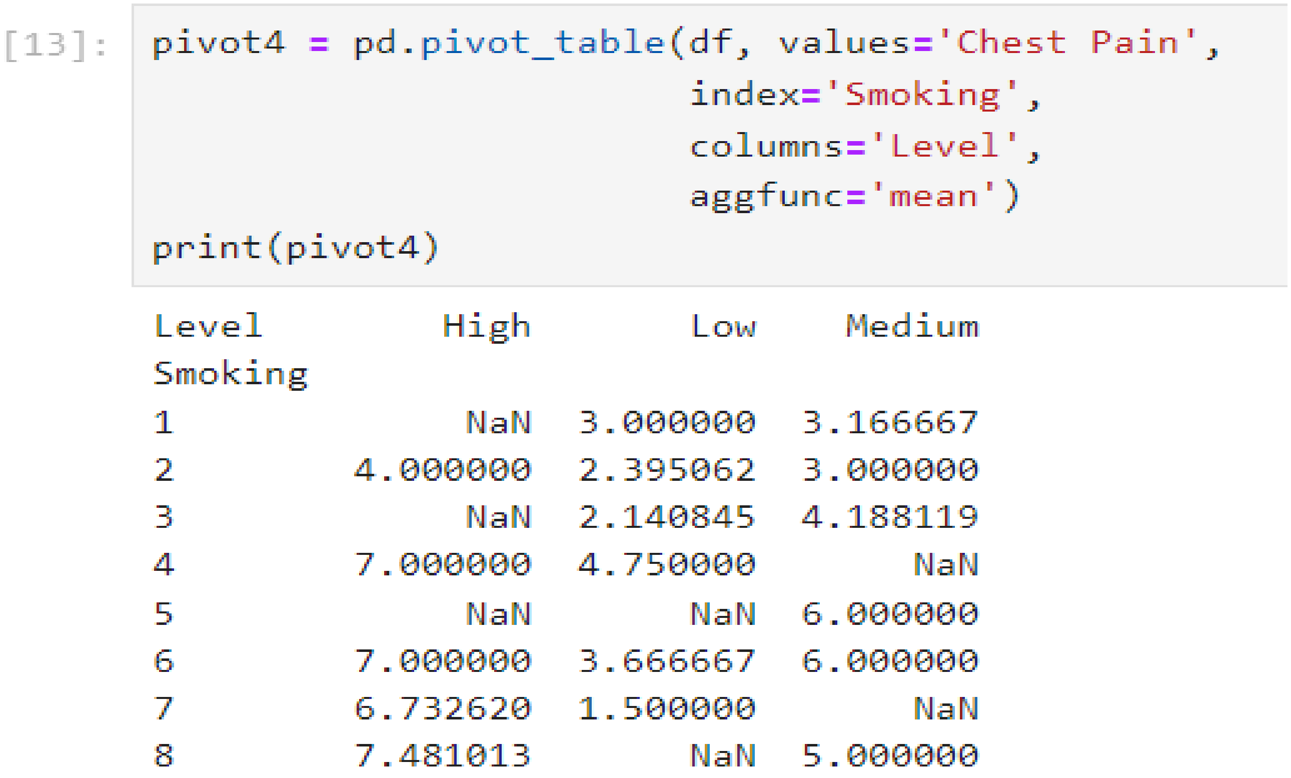
```
[11]: pivot3 = pd.pivot_table(df, values='Dust Allergy',
                             index='Level',
                             columns='Obesity',
                             aggfunc='mean')
      print(pivot3)
```

```
Obesity         1    2         3         4    5    6         7
Level
High          NaN  NaN  7.000000  2.000000  NaN  5.0  6.785276
Low      2.142857  3.2  3.444444  3.688525  NaN  NaN       NaN
Medium        NaN  6.5  3.598039  6.166667  6.0  5.5  7.000000
```



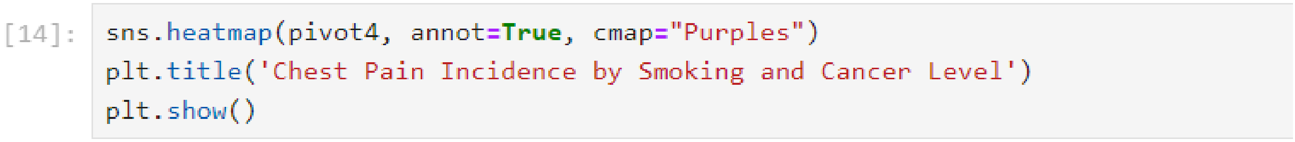
```
[12]: sns.heatmap(pivot3, annot=True, cmap="Oranges")
      plt.title('Frequency of Dust Allergy by Lung Cancer Level and Obesity')
      plt.show()
```



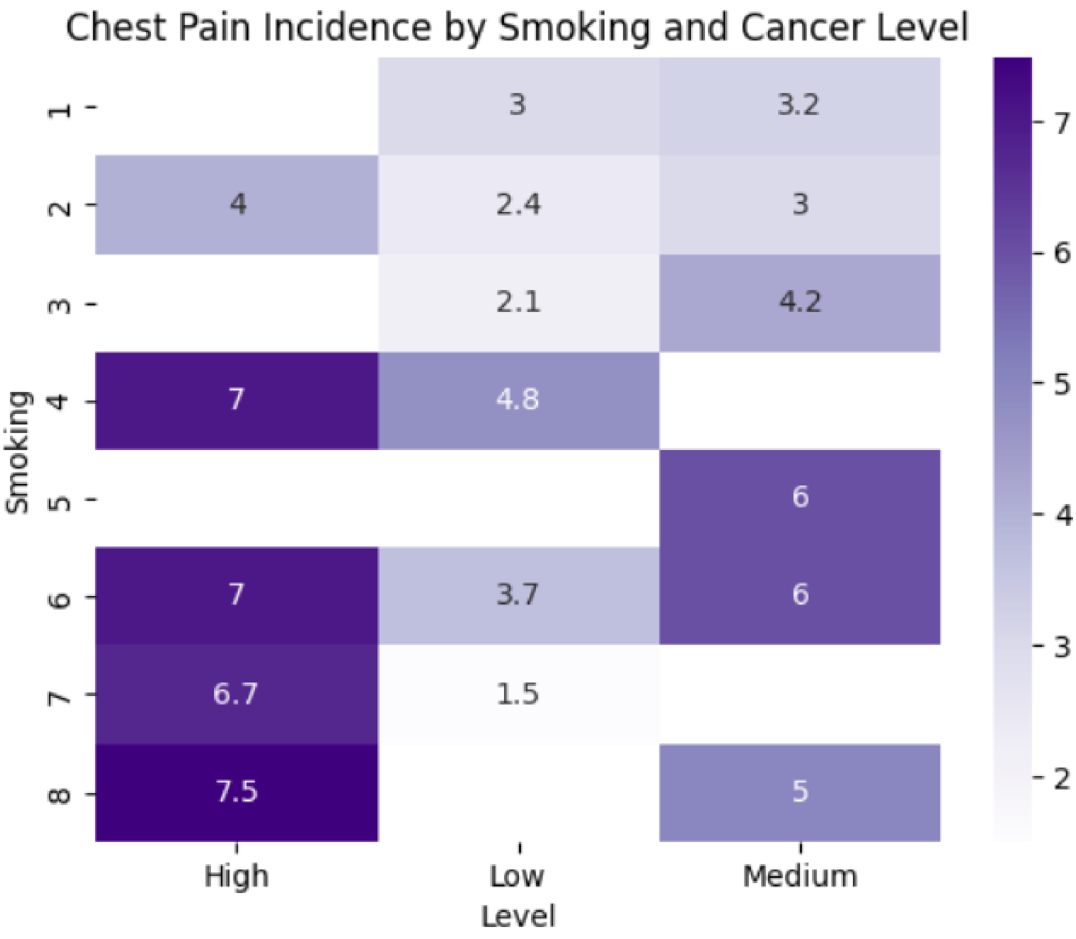


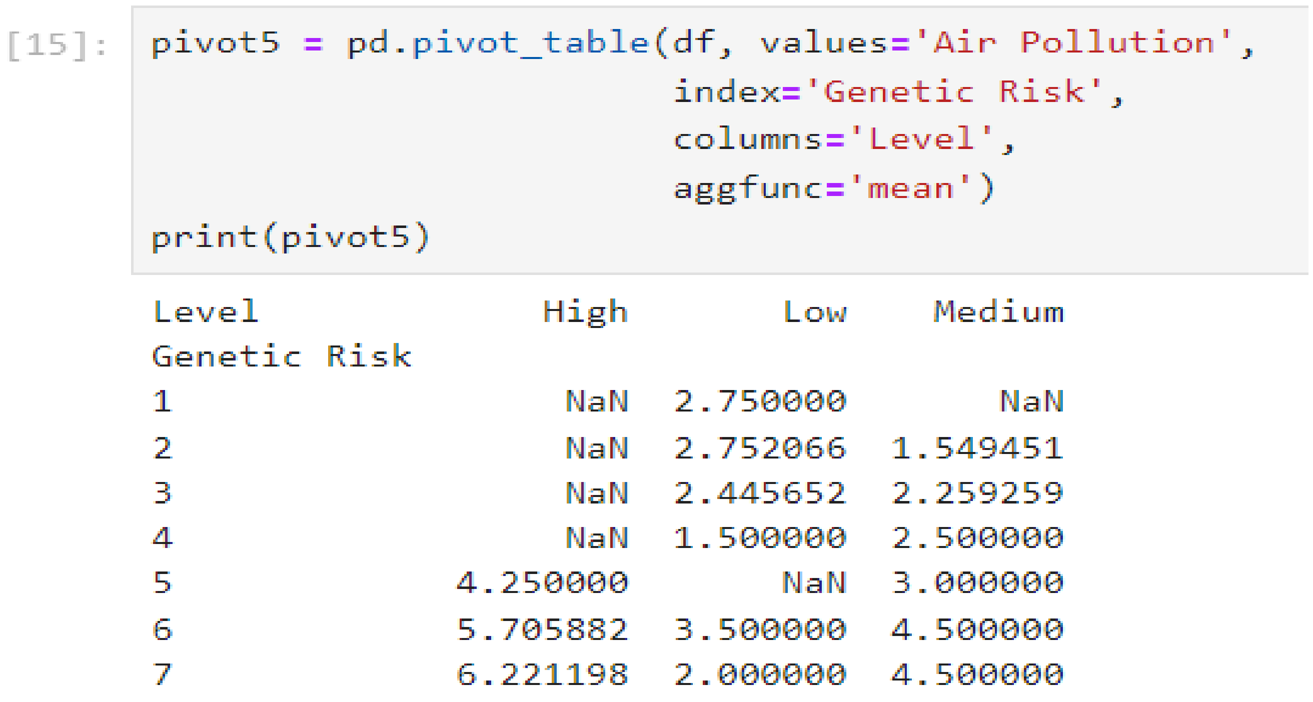
```
[13]: pivot4 = pd.pivot_table(df, values='Chest Pain',
                              index='Smoking',
                              columns='Level',
                              aggfunc='mean')
      print(pivot4)
```

```
Level        High       Low    Medium
Smoking
1             NaN  3.000000  3.166667
2        4.000000  2.395062  3.000000
3             NaN  2.140845  4.188119
4        7.000000  4.750000       NaN
5             NaN       NaN  6.000000
6        7.000000  3.666667  6.000000
7        6.732620  1.500000       NaN
8        7.481013       NaN  5.000000
```



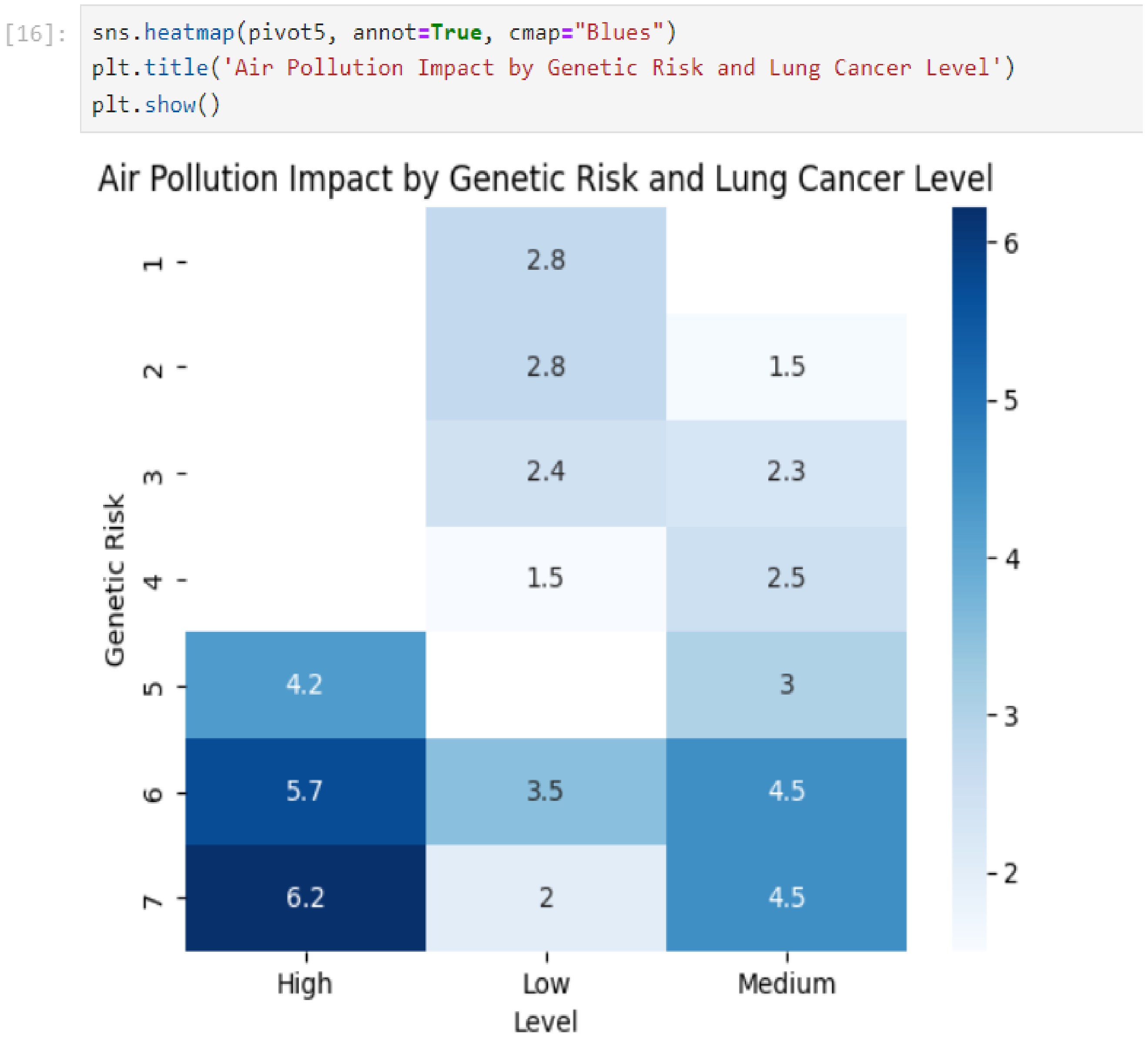
```
[14]: sns.heatmap(pivot4, annot=True, cmap="Purples")
      plt.title('Chest Pain Incidence by Smoking and Cancer Level')
      plt.show()
```



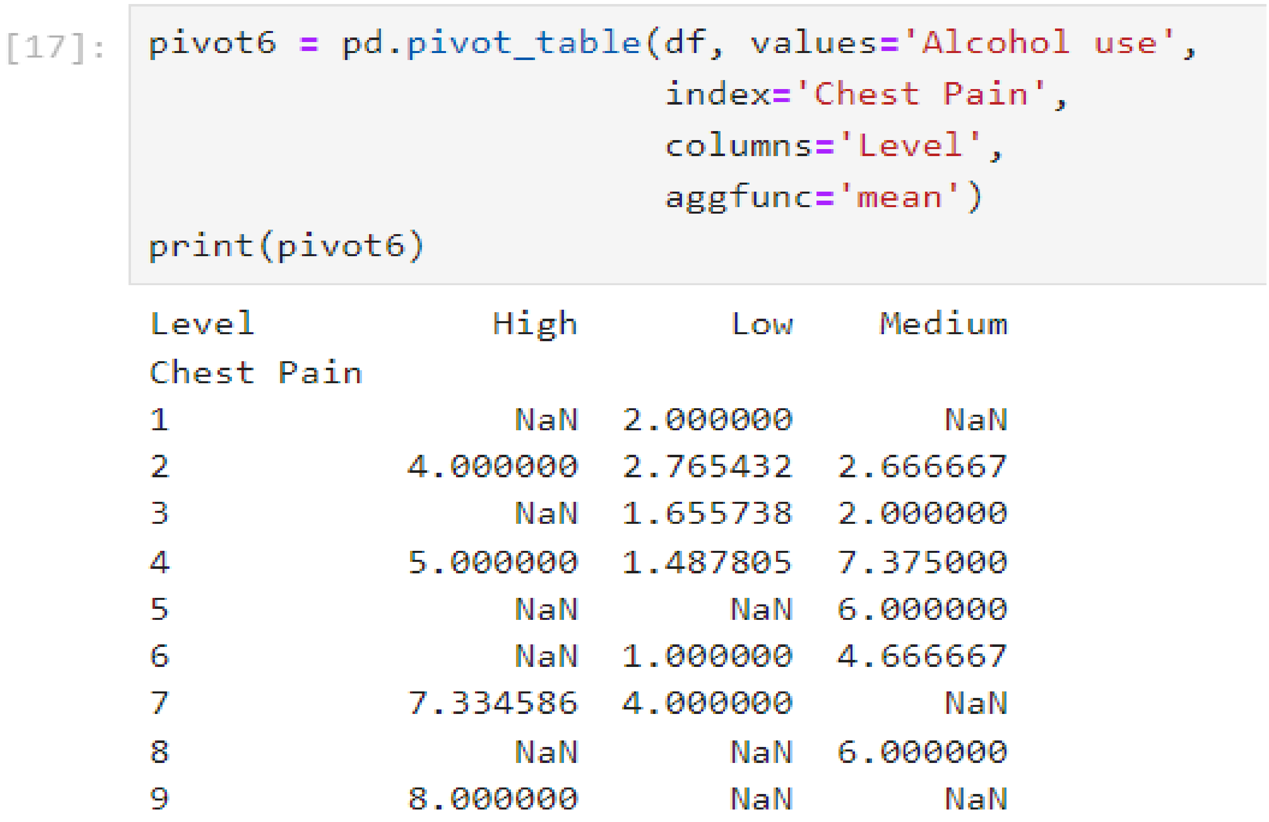


```
[15]: pivot5 = pd.pivot_table(df, values='Air Pollution',
                            index='Genetic Risk',
                            columns='Level',
                            aggfunc='mean')
      print(pivot5)

      Level               High       Low    Medium
      Genetic Risk
      1                    NaN  2.750000       NaN
      2                    NaN  2.752066  1.549451
      3                    NaN  2.445652  2.259259
      4                    NaN  1.500000  2.500000
      5               4.250000       NaN  3.000000
      6               5.705882  3.500000  4.500000
      7               6.221198  2.000000  4.500000
```

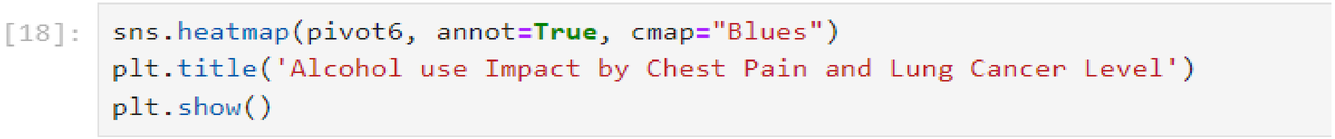


```
[16]: sns.heatmap(pivot5, annot=True, cmap="Blues")
      plt.title('Air Pollution Impact by Genetic Risk and Lung Cancer Level')
      plt.show()
```

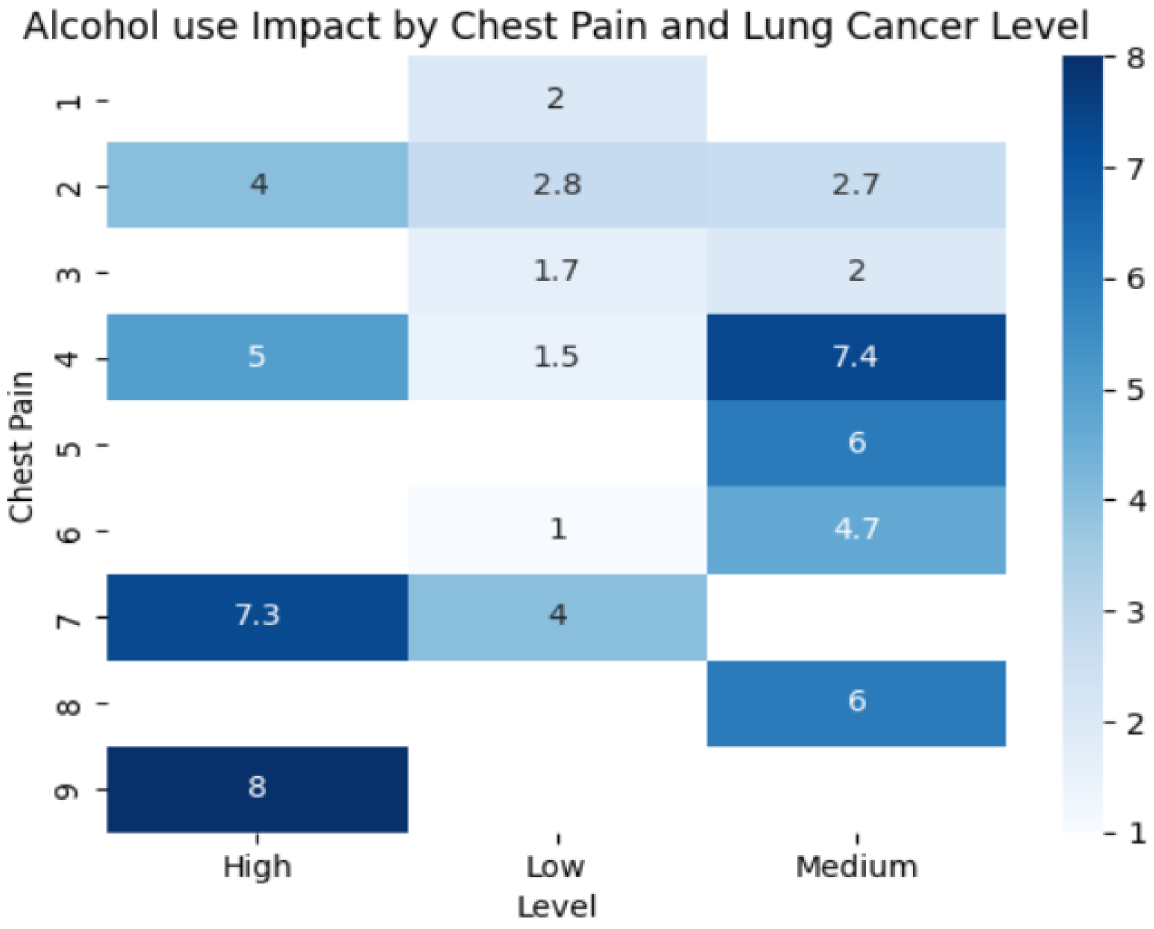


```
[17]: pivot6 = pd.pivot_table(df, values='Alcohol use',
                               index='Chest Pain',
                               columns='Level',
                               aggfunc='mean')
      print(pivot6)
```

```
Level            High       Low    Medium
Chest Pain
1                 NaN  2.000000       NaN
2            4.000000  2.765432  2.666667
3                 NaN  1.655738  2.000000
4            5.000000  1.487805  7.375000
5                 NaN       NaN  6.000000
6                 NaN  1.000000  4.666667
7            7.334586  4.000000       NaN
8                 NaN       NaN  6.000000
9            8.000000       NaN       NaN
```

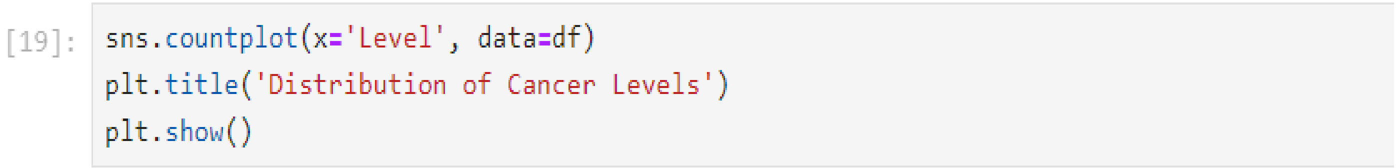


```
[18]: sns.heatmap(pivot6, annot=True, cmap="Blues")
      plt.title('Alcohol use Impact by Chest Pain and Lung Cancer Level')
      plt.show()
```

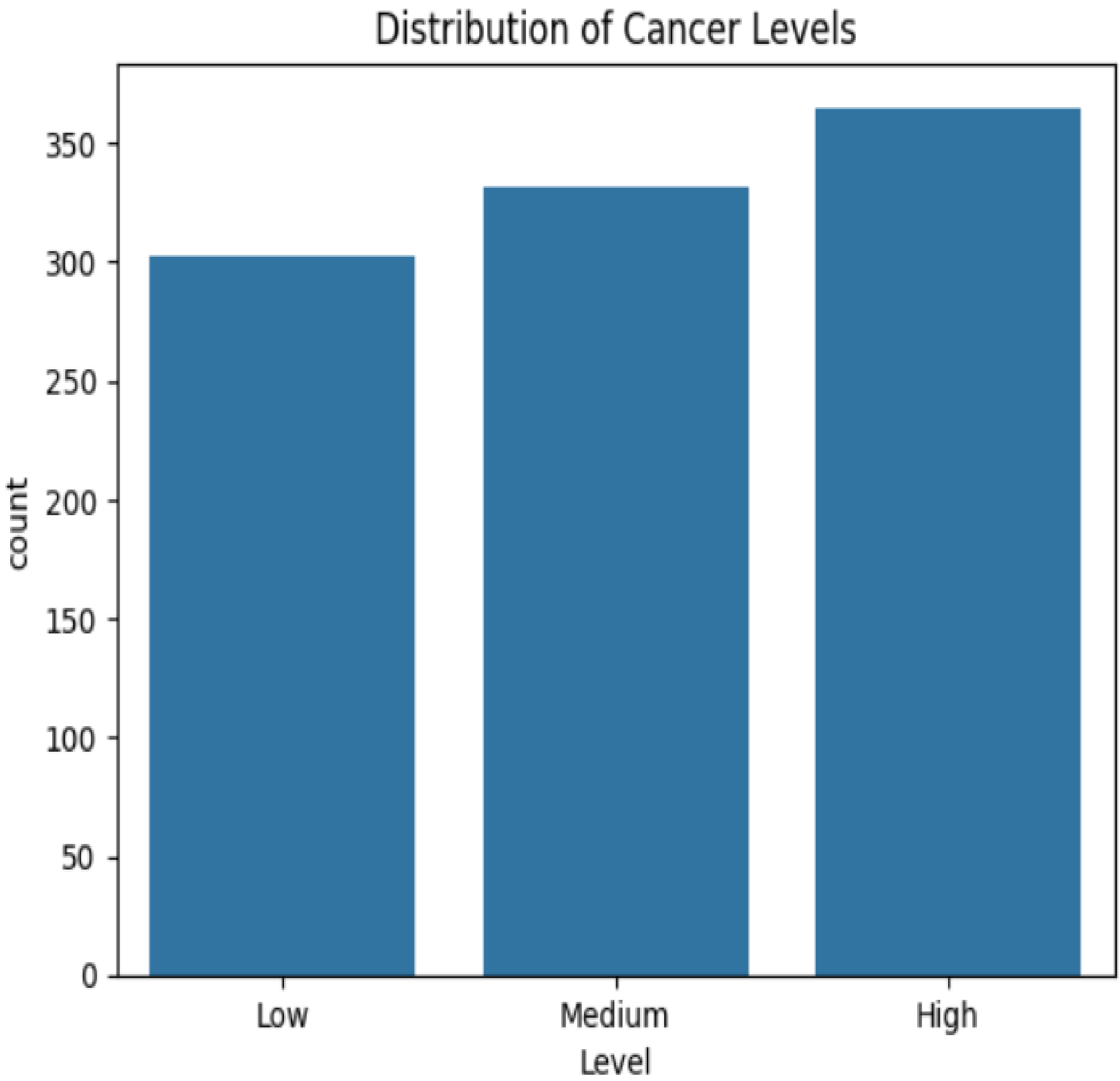


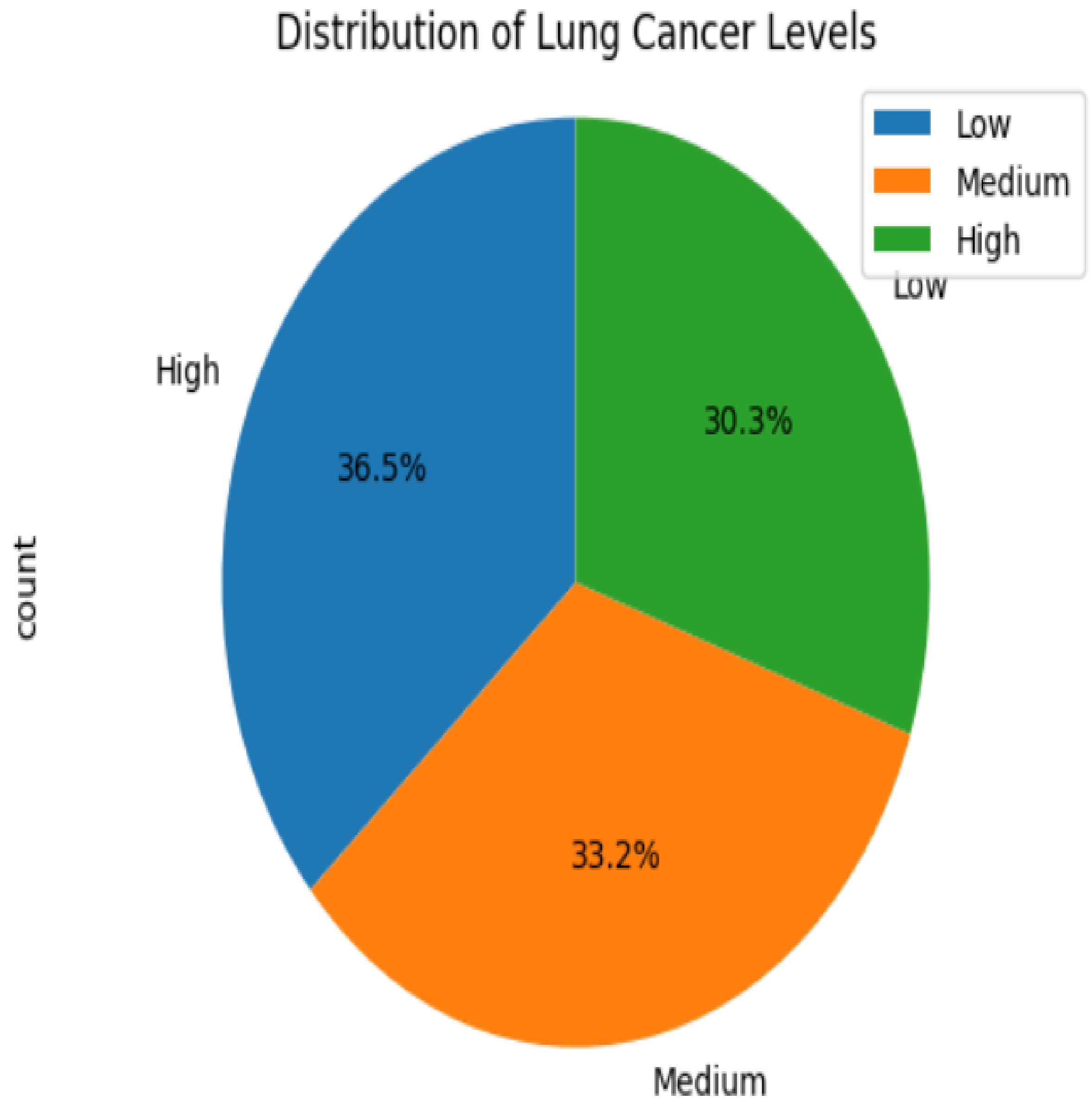
2. **Correlation Matrix:**
   - A correlation matrix was plotted using sns.heatmap() to check the relationships between different features. Features with high correlations are often more influential in the prediction process.
3. **Class Distribution:**
   - The dataset's target variable (cancer or no cancer) was checked for class imbalance using bar charts. This helped in determining whether techniques like SMOTE (Synthetic Minority Oversampling Technique) might be necessary.

Example Table:



```
[19]: sns.countplot(x='Level', data=df)
      plt.title('Distribution of Cancer Levels')
      plt.show()
```





4. Feature Selection and Engineering

Feature Selection:

After performing EDA, we selected features that exhibited high correlation with the target variable or had clinical significance, such as age, smoking habits, and family history of cancer.

Feature Engineering:

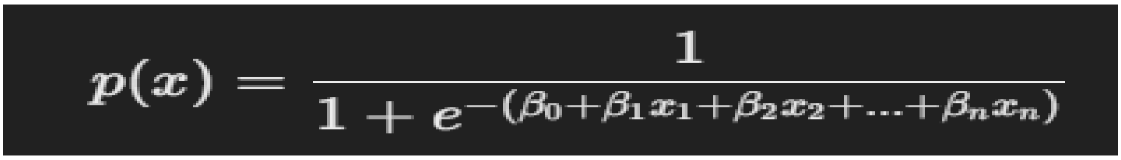
In some cases, new features were created based on existing ones, such as the creation of a "Risk Score" variable by combining smoking habits, family history, and age to better capture the likelihood of cancer occurrence.

5. Model Building

Models Used in This Project:

1. **Logistic Regression:**
   - Logistic regression was employed as it is a widely used algorithm for binary classification tasks.

*Formula for **Logistic Regression**:*



$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n)}}$$

The logistic function outputs probabilities, and a threshold of 0.5 was used to classify a patient as cancer-positive or negative.

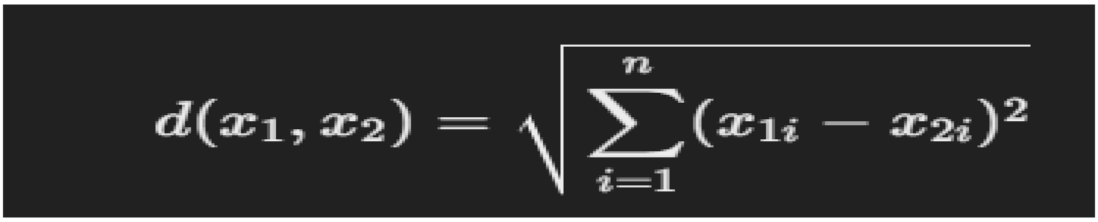
2. **Decision Tree Classifier:**
    - A decision tree classifier was used for its simplicity and interpretability. It works by recursively splitting the dataset based on feature values to arrive at a decision.

Key Parameters:

- max_depth: Controls the depth of the tree to avoid overfitting.
- criterion: Gini impurity was used to measure the quality of splits.

3. **K-Nearest Neighbors (KNN):**
    - KNN was applied as a non-parametric, instance-based learning algorithm that classifies based on the distance between the test sample and its nearest neighbors.

Key Formula for Distance:



$$d(x_1, x_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}$$

Here, the Euclidean distance was used to find the closest points in the feature space.

Model Performance:

- After training each model, we evaluated their performance using metrics such as accuracy, precision, recall, F1-score, and AUC (Area Under the ROC Curve).

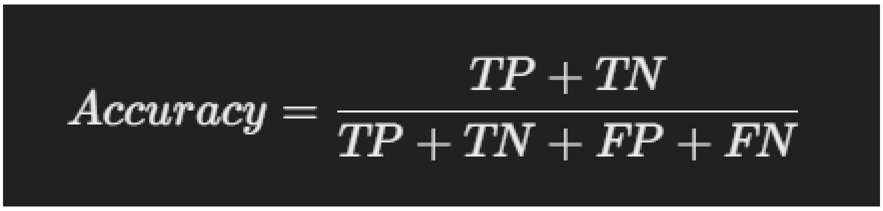
Example Comparison Table:

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Logistic Regression	100%	1.00	1.00	1.00
Decision Tree	100%	1.00	1.00	1.00
K-Nearest Neighbors	80%	0.81	0.92	0.86

6. Model Evaluation

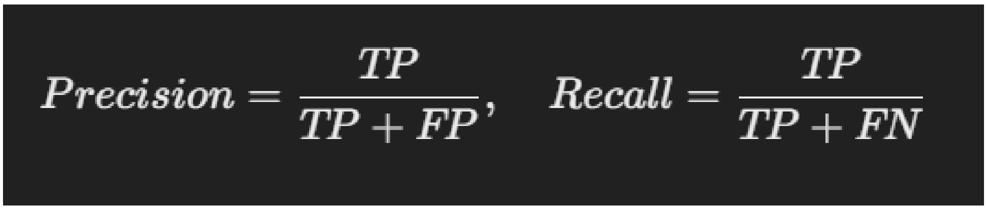
Evaluation Metrics:

1. **Accuracy:**
   - The percentage of correctly classified instances.
   - Formula:



$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2. **Precision & Recall:**
   - Precision measures how many of the positively classified instances are truly positive.
   - Recall (Sensitivity) measures how many actual positive instances are correctly identified by the model.
   - Formulas:



$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN}$$

3. **F1-Score:**
   - The harmonic mean of precision and recall, providing a balanced measure even if class distribution is uneven.
4. **ROC Curve & AUC:**
   - The ROC curve shows the trade-off between true positive rate and false positive rate at different thresholds, and the AUC indicates the overall performance of the model.

7. Conclusion

- Logistic Regression emerged as the most accurate model for predicting lung cancer, with an accuracy of 82% and a strong AUC score.
- The project demonstrates the potential of machine learning in healthcare for predictive diagnostics.
- The EDA revealed significant insights into the data, such as the high correlation between smoking and lung cancer, which influenced feature selection and model building.
- Future improvements could include collecting more diverse data and applying advanced techniques like ensemble methods (Random Forest, Gradient Boosting) for better accuracy.