The goal of this project will be to clean the dataset, if necessary, perform Exploratory Data Analysis, and build a logistic regression model and evaluate its performance

Link: https://www.kaggle.com/datasets/thedevastator/cancer-patients-and-air-pollution-a-new-link

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [2]:
```
data = pd.read_csv('cancer_patient_data_sets.csv')
```

In [3]:
```
pd.set_option('display.max_columns', None)
data.head()
```

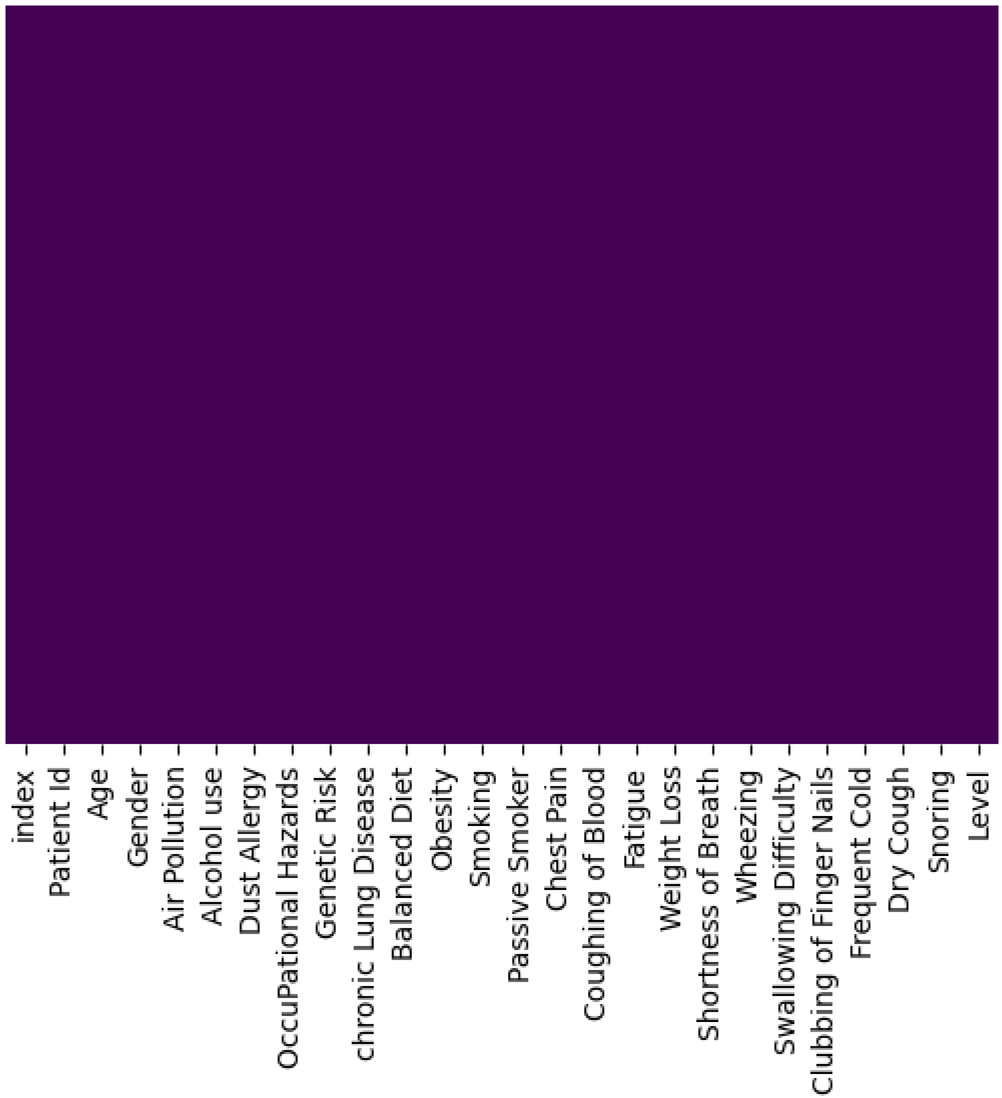
Out[3]:

	index	Patient Id	Age	Gender	Air Pollution	Alcohol use	Dust Allergy	OccuPational Hazards	Genetic Risk	chronic Lung Disease	Bala
0	0	P1	33	1	2	4	5	4	3	2	
1	1	P10	17	1	3	1	5	3	4	2	
2	2	P100	35	1	4	5	6	5	5	4	
3	3	P1000	37	1	7	7	7	7	6	7	
4	4	P101	46	1	6	8	7	7	7	6	

Checking for Missing Data

In [5]:
```
sns.heatmap(data.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[5]:
```
<AxesSubplot: >
```



There are no missing values in any of the columns. Before proceeding with Logistic Regression, the 'index' and 'Patient Id' columns will be dropped since they are not needed for the Logistic Regression problem.

In [6]:
```
data.drop(['index','Patient Id'], axis=1, inplace=True)
```

In [7]:
```
data.head()
```

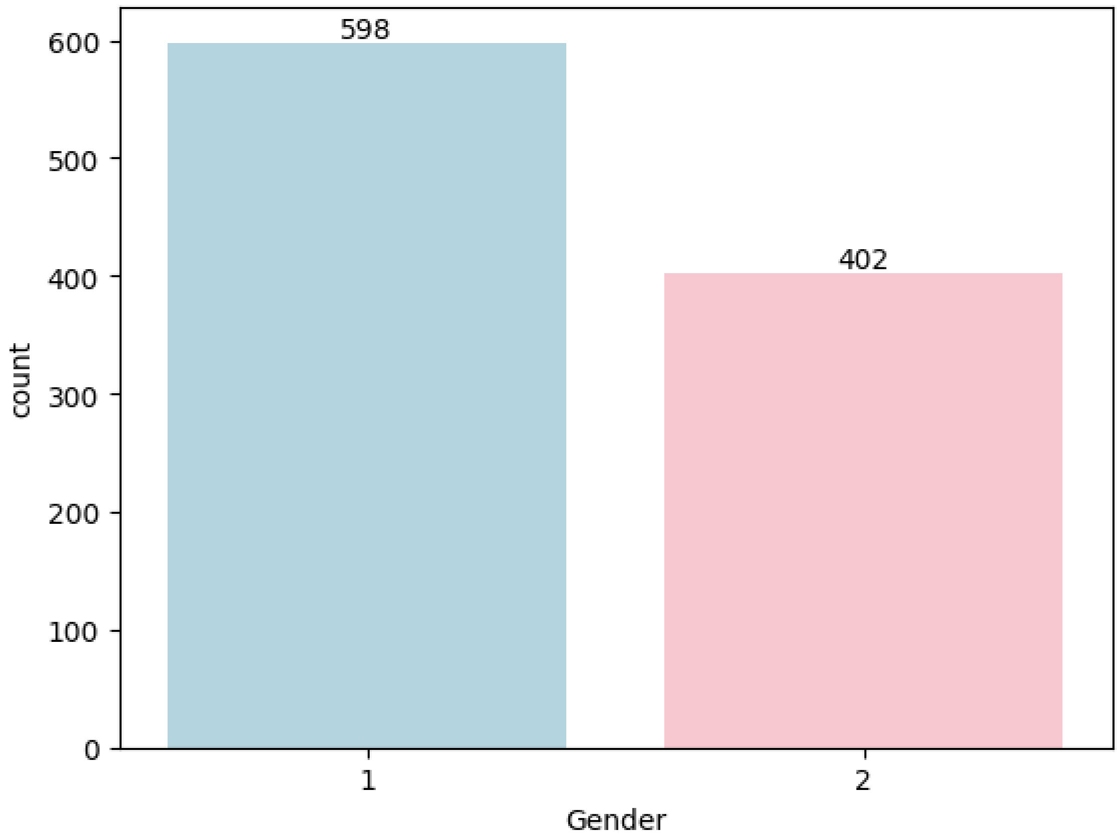
Out[7]:

	Age	Gender	Air Pollution	Alcohol use	Dust Allergy	OccuPational Hazards	Genetic Risk	chronic Lung Disease	Balanced Diet	Obesity
0	33	1	2	4	5	4	3	2	2	4
1	17	1	3	1	5	3	4	2	2	2
2	35	1	4	5	6	5	5	4	6	7
3	37	1	7	7	7	7	6	7	7	7
4	46	1	6	8	7	7	7	6	7	7

Exploratory Data Analysis (EDA)

In [8]:
```
ax = sns.countplot(x=data['Gender'], palette=['LightBlue','Pink'])
ax.bar_label(ax.containers[0])
```

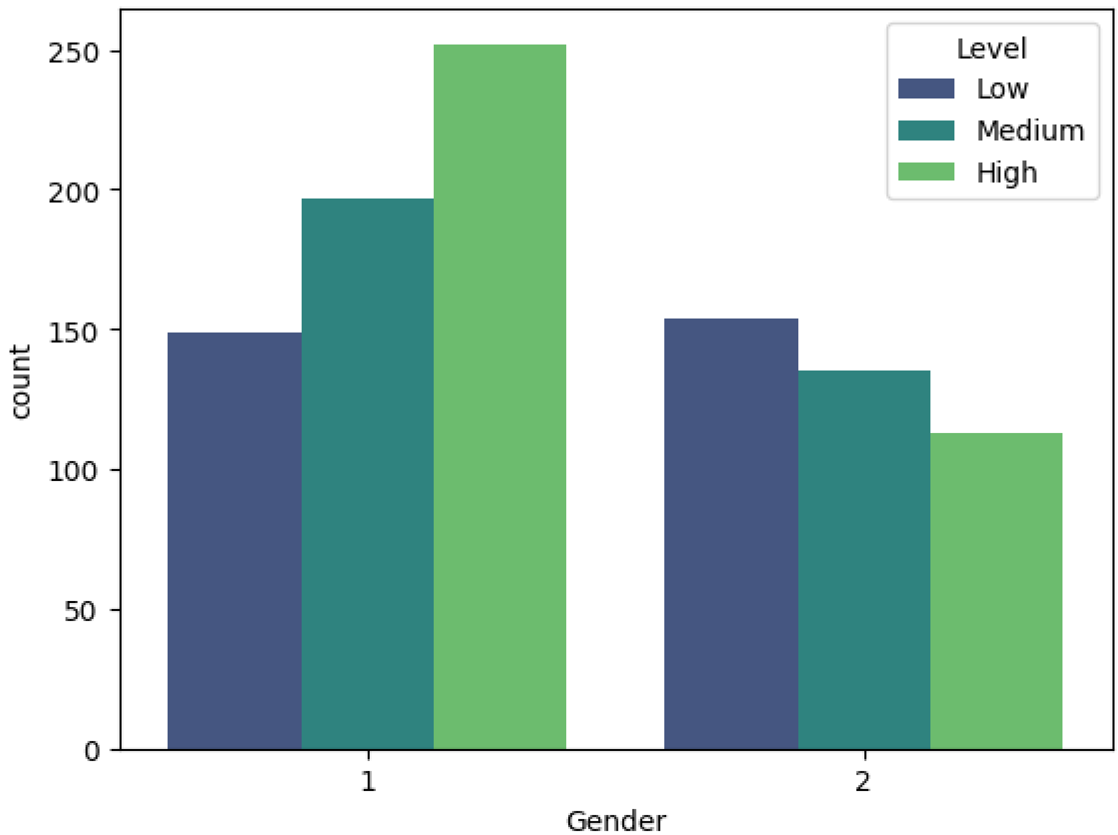
Out[8]: [Text(0, 0, '598'), Text(0, 0, '402')]



In the dataset we see that there are more males than females that have cancer.

In [9]:
```
sns.countplot(x=data['Gender'],hue=data['Level'],palette='viridis')
```

Out[9]: <AxesSubplot: xlabel='Gender', ylabel='count'>

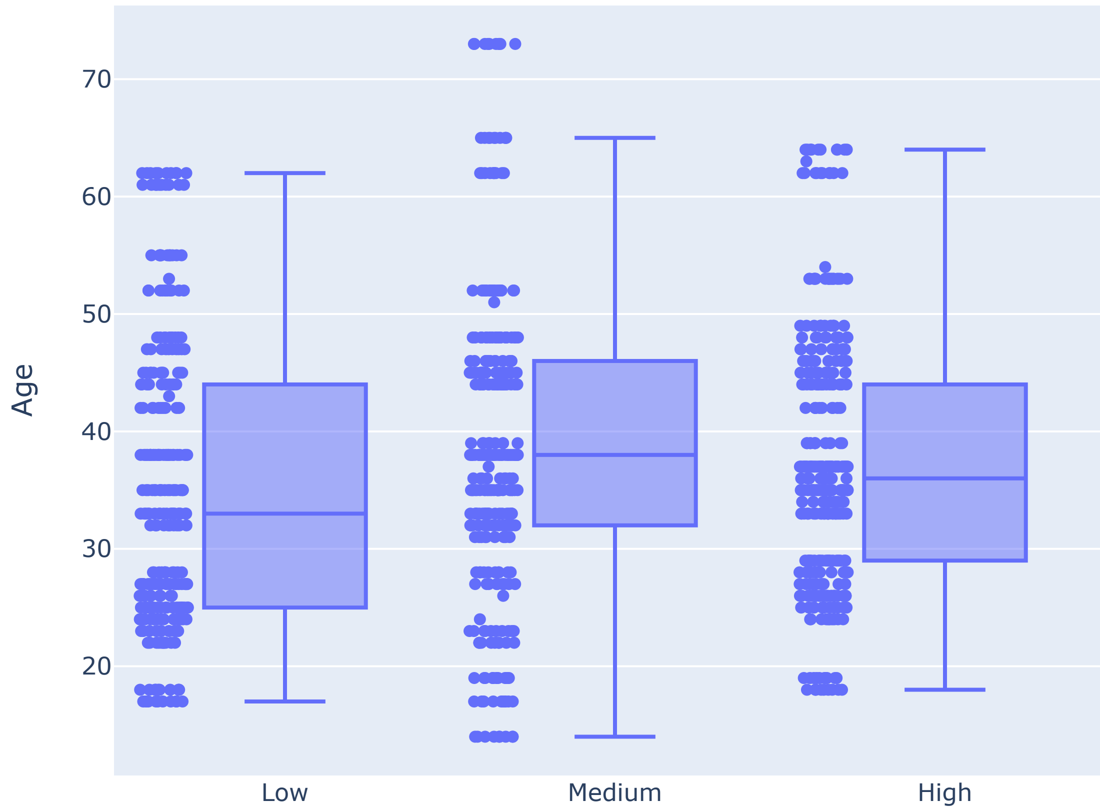


Within the men group, we see that they are more likely to have a higher level of lung cancer. Within the women group, women are more likely to have a lower level of lung cancer than men. Moreover, en are more likely to have a higher level of lung cancer than women.

In [ ]:
```
import cufflinks as cf
cf.go_offline()
```

In [10]:
```
import plotly.express as px
```

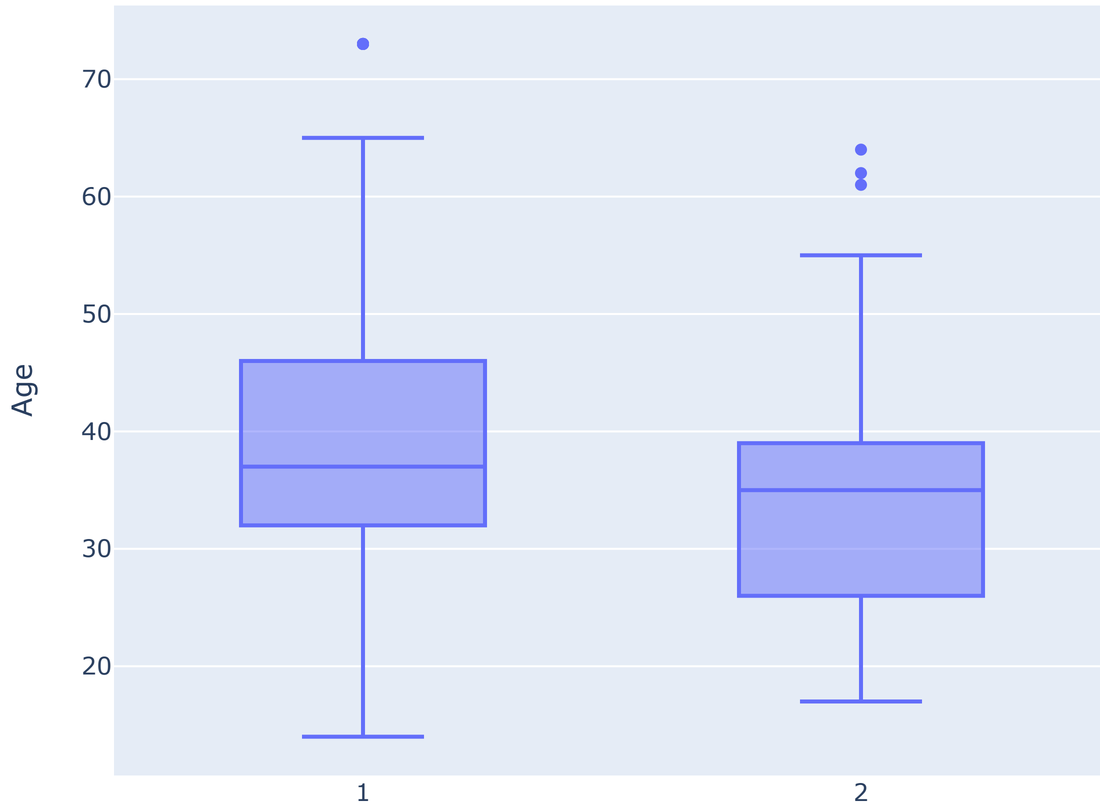
In [11]:
```
fig = px.box(data, x="Level", y="Age", points='all',)
fig.show()
```



We see from the boxplots in the above graph that the levels have roughly similar ranges with regards to age. However, we do see that the Medium level has a larger range than the other two levels. The median age's for the Low, Medium, and High levels are 33, 38, and 36, respectively. The box for the level, low, is larger than the other two levels, thus, most of the values are spread out from the median.

Creating box plots displaing the age range for each gender.

In [12]:
```
fig = px.box(data, x="Gender", y="Age")
fig.show()
```

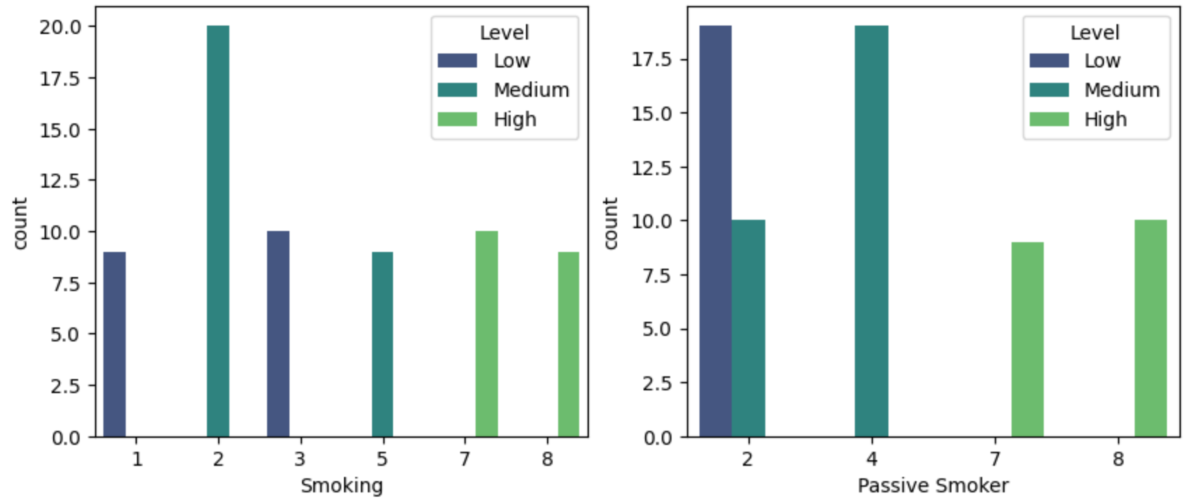


Based on the boxplots, 75% of lung cancer patients who are men are between the ages 14 and 46. 75% of women lung cancer patients are between the ages 17-39. The men's age group has larger age distribution than the women's.

Since there are patients that are under the age 21, what will be looked at next is to see patients that are under the age of 21, who are actively and passively smoking to see what level of lung cancer they have. (As of 2022, individuals must be 21 years or older to purchase cigarretes).

In [13]:
```
fig, axes = plt.subplots(1, 2, figsize=(10,4))

sns.countplot(x = data[data['Age']<21]['Smoking'], palette='viridis',hue=data['Leve


sns.countplot(x = data[data['Age']<21]['Passive Smoker'], palette='viridis',hue=dat
```

Out[13]: <AxesSubplot: xlabel='Passive Smoker', ylabel='count'>



The goal of the visualization was to see patients under 21 years of age that were smoking, either passively or actively, what their level of lung cancer would be based on the level of smoking. It is not suprising to see that the higher level of smokers had a high level of lung cancer.

Logistic Regression

In [14]:
```
from sklearn.model_selection import train_test_split
```

In [15]:
```
X = data.drop('Level', axis=1)

y = data['Level']
```

In [16]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_st
```

In [17]:
```
from sklearn.linear_model import LogisticRegression
```

In [21]:
```
#Training the Model
logmodel = LogisticRegression(max_iter=3000)
logmodel.fit(X_train,y_train)
```

Out[21]:
```
▾ LogisticRegression
LogisticRegression(max_iter=3000)
```

In [22]:
```
#Using the model to predict
predictions = logmodel.predict(X_test)
```

In [23]:
```
from sklearn.metrics import classification_report, confusion_matrix, ConfusionMatri
```

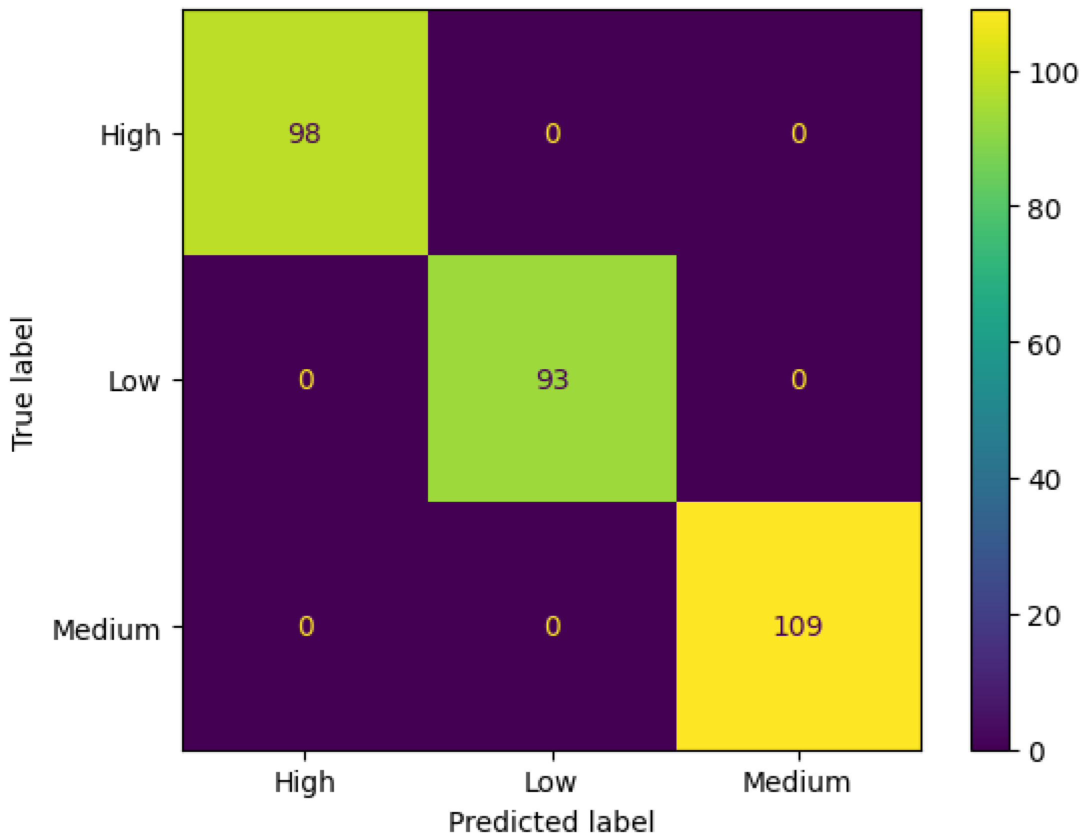
In [24]:
```
#Displaying the classification report
print(classification_report(y_test, predictions))
```

```
              precision    recall  f1-score   support

        High       1.00      1.00      1.00        98
         Low       1.00      1.00      1.00        93
      Medium       1.00      1.00      1.00       109

    accuracy                           1.00       300
   macro avg       1.00      1.00      1.00       300
weighted avg       1.00      1.00      1.00       300
```

In [25]:
```
print(confusion_matrix(y_test,predictions))
```

```
[[ 98   0   0]
 [  0  93   0]
 [  0   0 109]]
```

In [26]:
```
#Confusion Matrix in Graph Form
ConfusionMatrixDisplay.from_predictions(y_test, predictions)
plt.show()
```



The precision and recall scores all come back at a high level. The same thing can be said for the F1-Score. Thus, the logistic regression model has done well at predicting the correct values and is fit to be deployed in the real-world.