

```
In [1]: import pandas as pd
        from sklearn.preprocessing import LabelEncoder, StandardScaler
```

```
In [3]: df = pd.read_csv("Downloads/medical_dataset.csv")
```

```
In [5]: encoder = LabelEncoder()
        df["Gender"] = encoder.fit_transform(df["Gender"])
```

```
In [7]: scaler = StandardScaler()
        df[["Age"]] = scaler.fit_transform(df[["Age"]])
```

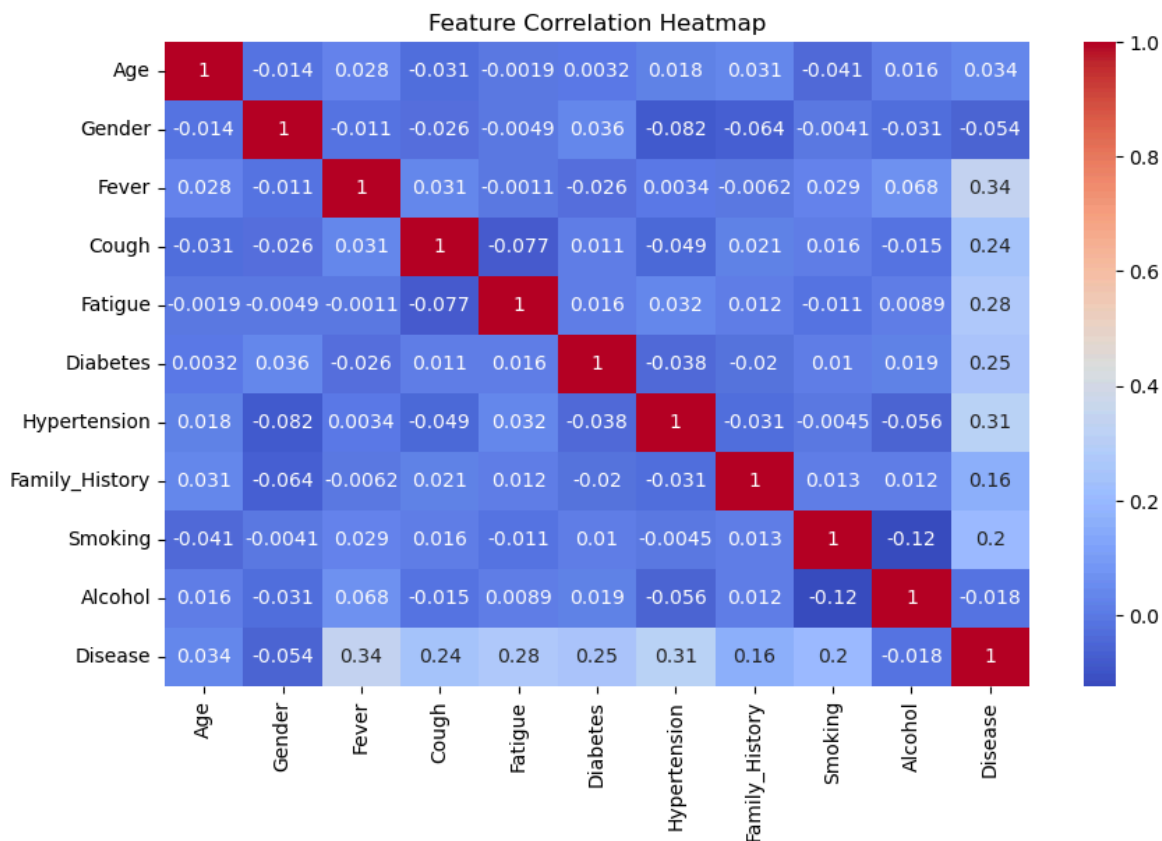
```
In [9]: print(df.head())
```

	Age	Gender	Fever	Cough	Fatigue	Diabetes	Hypertension	\
0	0.449200	0	0	0	0	0		0
1	1.197866	0	1	0	1	0		0
2	-0.126697	1	0	1	0	0		0
3	-0.932953	0	0	0	1	0		0
4	0.679558	1	1	1	0	0		0

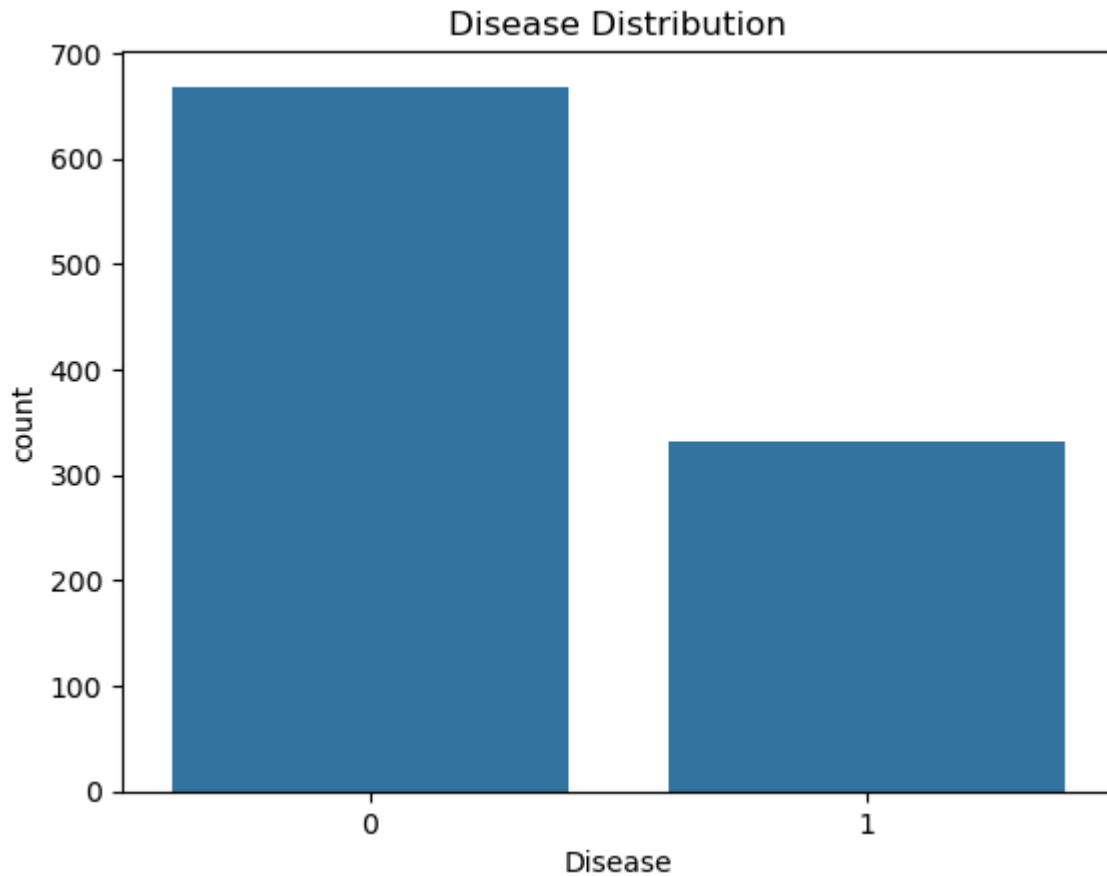
	Family_History	Smoking	Alcohol	Disease
0	0	0	1	0
1	1	0	1	0
2	0	1	0	0
3	0	1	0	0
4	0	0	1	1

```
In [11]: import seaborn as sns
          import matplotlib.pyplot as plt
```

```
In [13]: plt.figure(figsize=(10, 6))
          sns.heatmap(df.corr(), annot=True, cmap="coolwarm")
          plt.title("Feature Correlation Heatmap")
          plt.show()
```



```
In [15]: sns.countplot(x="Disease", data=df)
plt.title("Disease Distribution")
plt.show()
```



```
In [17]: from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
```

```
from sklearn.metrics import accuracy_score, classification_report
```

```
In [19]: X = df.drop("Disease", axis=1) # Features
y = df["Disease"] # Target variable
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_
```

```
In [21]: model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

```
Out[21]: ▼ RandomForestClassifier ⓘ ?
RandomForestClassifier(random_state=42)
```

```
In [23]: y_pred = model.predict(X_test)
```

```
In [25]: print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

Accuracy: 1.0

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	130
1	1.00	1.00	1.00	70
accuracy			1.00	200
macro avg	1.00	1.00	1.00	200
weighted avg	1.00	1.00	1.00	200

```
In [27]: from sklearn.model_selection import GridSearchCV

param_grid = {
    "n_estimators": [50, 100, 150],
    "max_depth": [5, 10, 15],
}

grid_search = GridSearchCV(RandomForestClassifier(), param_grid, cv=5)
grid_search.fit(X_train, y_train)

print("Best Parameters:", grid_search.best_params_)
```

Best Parameters: {'max_depth': 10, 'n_estimators': 100}

```
In [29]: new_patient = [[45, 1, 1, 0, 1, 0, 1, 0, 0, 1]]
prediction = model.predict(new_patient)
print("Disease Prediction:", "Yes" if prediction[0] == 1 else "No")
```

Disease Prediction: Yes

C:\Users\HP\anaconda3\Lib\site-packages\sklearn\base.py:493: UserWarning: X does not have valid feature names, but RandomForestClassifier was fitted with feature names
warnings.warn(

```
In [ ]:
```