Predicting Chronic Obstructive Pulmonary Disease (COPD) in Nepal Project

The approaches mentioned below were followed to develop the system for COPD, including data collection, preprocessing, feature engineering, model development, and deployment.

Step 1: Problem Statement and Objectives

Problem Statement:

- To predict the possibility of a patient developing COPD based on various risk factors and patient characteristics.

Objectives:

1. Collect and preprocess data relevant to COPD.
2. Identify and engineer significant features contributing to COPD.
3. Develop a predictive model to estimate the risk of COPD.
4. Evaluate the model's performance and refine it.
5. Deploy the model for practical use in a clinical or public health setting.

Step 2: Data Collection

Various sites were visited and research was conducted to find good datasets, but all of them did not have a target variable: whether the patient has COPD or doesn't have COPD, so the "***synthetic_COPD_data.csv***" was used for the project.

Data sources researched and identified

- Health Data Portals: WHO, World Bank, Nepal's Ministry of Health
- Research Papers: Google Scholar
- Public Datasets: Kaggle, Data.gov, Open Data Nepal
- Hospitals and Clinics

Step 3: Data Preprocessing

The initially found data, i.e. ***"finalalldata.csv"*** and ***"Dataset_PowerBI.xlsx"*** were studied and merged as shown in the ***"data_prep.ipynb"*** file by reducing the columns and only using the necessary ones. But, later "***synthetic_COPD_data.csv***" was used for the project.

- Sample of ***finalalldata.csv***

```
csv_data.head()
```

Python

	uid	label	sex	age	bmi	smoke	location	rs10007052	rs8192288	rs20541	rs12922394	rs2910164	rs161976	rs473892	rs159497
0	copdcontrol1	0	2	28	19.22	0	4.63	1.671	1.0	0.448632	0.42328	1.000	1.0	1.473	1.000000
1	copdcontrol69	0	1	53	20.44	0	4.63	1.671	1.0	1.000000	0.65060	1.416	1.0	1.000	1.000000
2	copdcontrol68	0	1	58	20.45	1	4.63	1.000	1.0	0.669800	1.00000	1.416	1.0	1.473	1.000000
3	copdcontrol85	0	2	30	20.70	0	4.63	1.671	1.0	0.448632	1.00000	1.416	NaN	1.473	1.000000
4	copdcontrol78	0	1	55	20.76	1	4.63	1.671	1.0	0.669800	0.65060	1.000	1.0	1.473	2.088025

- Sample of ***"Dataset_PowerBI.xlsx"***

```
excel_data.head()
```

Python

	Patient ID	Gender	Age	Heigt(cm)	Height(in)	weight kg	weight (lb)	BMI	Smoker	Comorbidity1	...	RR During vigorous exercise	Baseline FEV1	Current FEV1	Baseline VO2	Re
0	3	Male	88	178	69.954	75	165.300	23.746689	No	NaN	...	74	0.729397	0,89 L	2.146	3.1
1	4	Female	60	167	65.631	79	174.116	28.416852	No	NaN	...	41	1.358489	1,179 L	1.780	13.1
2	8	Male	55	178	69.954	115	253.460	36.411591	No	Pulmonary Hypertension	...	47	1.588316	1,744 L	3.083	8.3
3	9	Female	55	170	66.810	81	178.524	28.117001	No	NaN	...	59	0.549915	0,526 L	1.963	5.0
4	10	Male	65	169	66.417	75	165.300	26.343269	No	NaN	...	42	1.337281	1,149 L	2.616	5.1

5 rows × 59 columns

- Sample of the merged dataset with relevant columns

	sex	bmi	age	location	smoke	Heigt(cm)	weight kg	BMI	Smoker	Gender
0	1	20.44	53	4.63	0	172	69	23.397745	Yes	Male
1	1	20.44	53	4.63	0	179	120	37.571367	No	Male
2	1	20.44	53	4.63	0	173	96	32.178133	Yes	Female
3	1	20.44	53	4.63	0	175	98	32.101978	No	Male
4	1	20.44	53	4.63	0	170	89	30.893988	Yes	Male

Step 4: Exploratory Data Analysis (EDA)

Then, EDA was performed on the datasets following the mentioned steps:

1. Loading the datasets and studying them
   Sample of the dataset used:

```python
#Load the datasets
path = r"../Data/synthetic_COPD_data.csv"
df = pd.read_csv(path)
df.head()
```

Python

	Age	Gender	Smoking_Status	Biomass_Fuel_Exposure	Occupational_Exposure	Family_History_COPD	BMI	Location	Air_Pollution_Level	Respira
0	31	Male	Former	1	1	1	27.56	Lalitpur	84	
1	60	Male	Never	1	0	0	30.30	Pokhara	131	
2	33	Male	Former	0	0	1	28.45	Pokhara	123	
3	36	Female	Current	1	0	0	27.49	Kathmandu	253	
4	58	Male	Never	0	0	0	25.49	Pokhara	117	

Then, a plot was set for better visibility and the data was studied along with the columns and values that were present using ***df.head()*** and ***df.info()***.

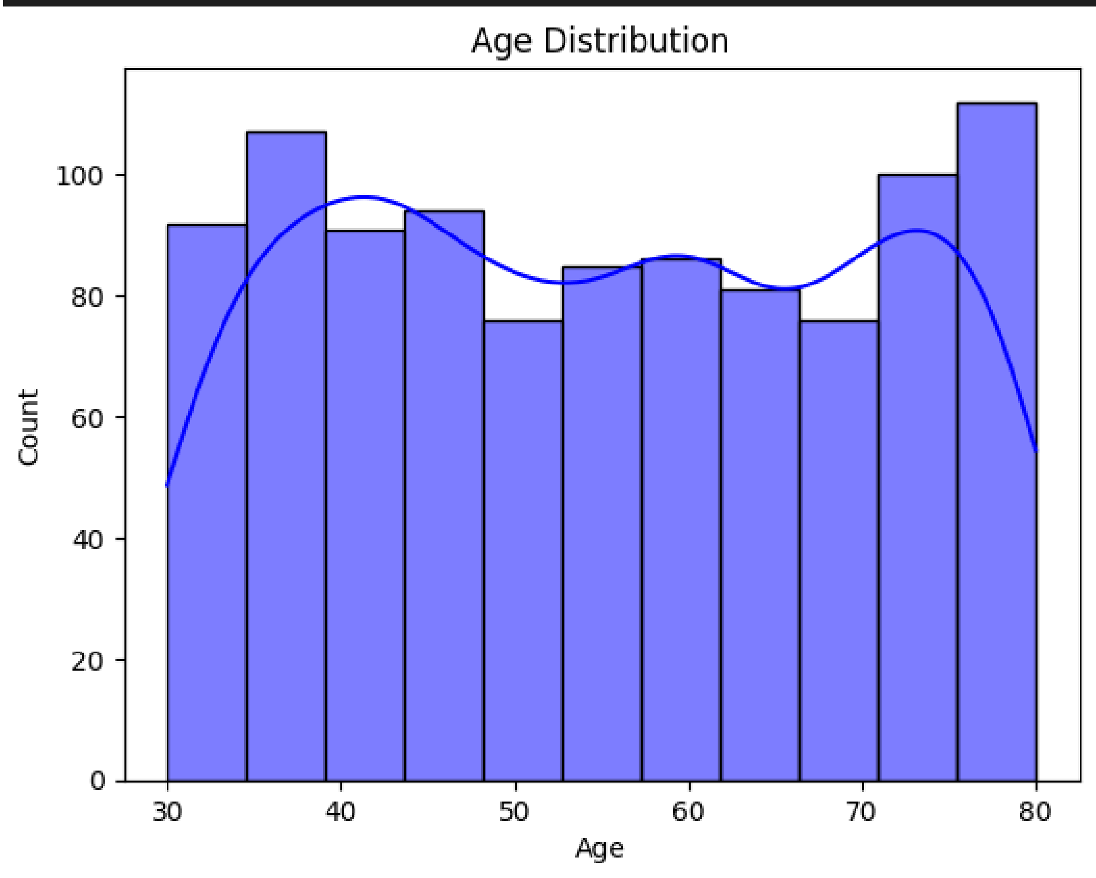
```python
#Get info about the data
df.info()
```

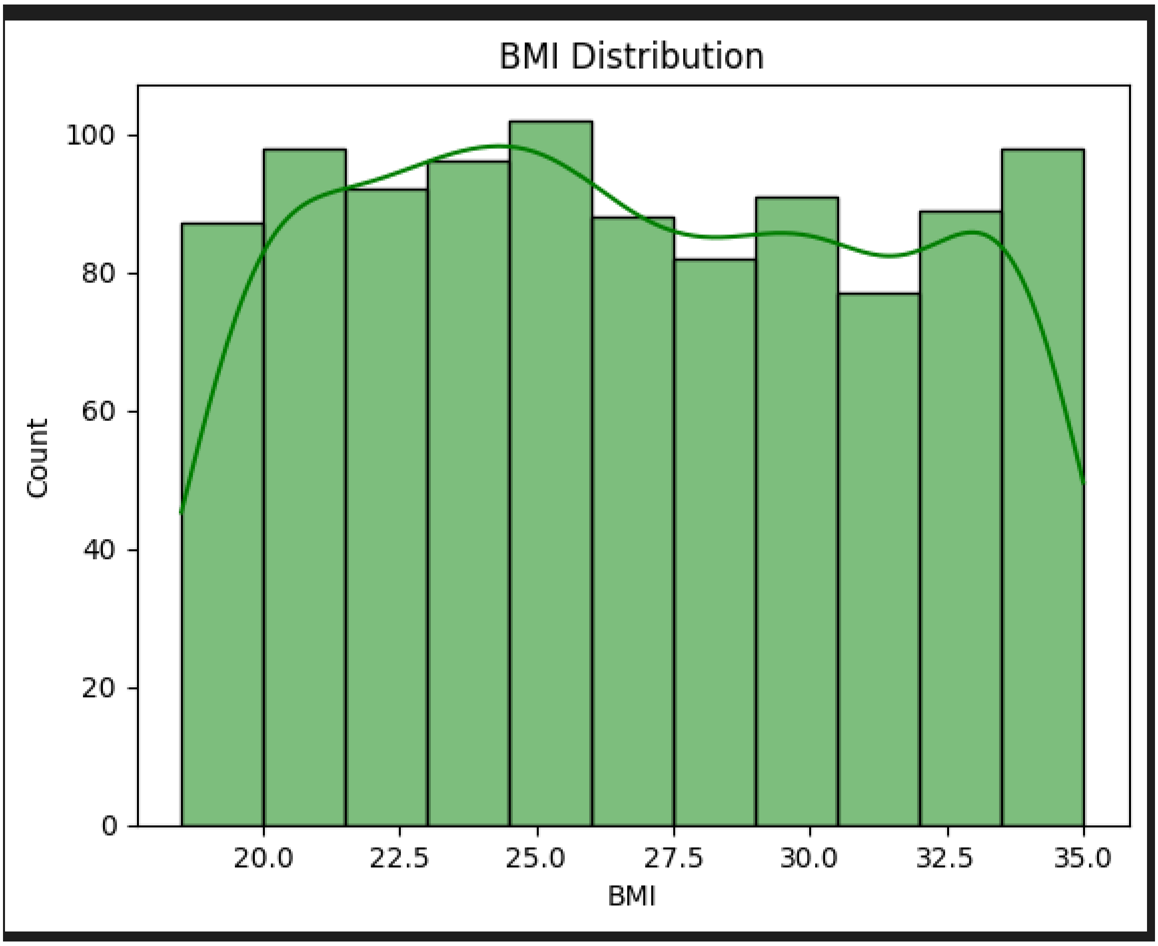
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 11 columns):
 #   Column                            Non-Null Count  Dtype  
---  ------                            --------------  -----  
 0   Age                               1000 non-null   int64  
 1   Gender                            1000 non-null   object 
 2   Smoking_Status                    1000 non-null   object 
 3   Biomass_Fuel_Exposure             1000 non-null   int64  
 4   Occupational_Exposure             1000 non-null   int64  
 5   Family_History_COPD               1000 non-null   int64  
 6   BMI                               1000 non-null   float64
 7   Location                          1000 non-null   object 
 8   Air_Pollution_Level               1000 non-null   int64  
 9   Respiratory_Infections_Childhood  1000 non-null   int64  
 10  COPD_Diagnosis                    1000 non-null   int64  
dtypes: float64(1), int64(7), object(3)
memory usage: 86.1+ KB
```

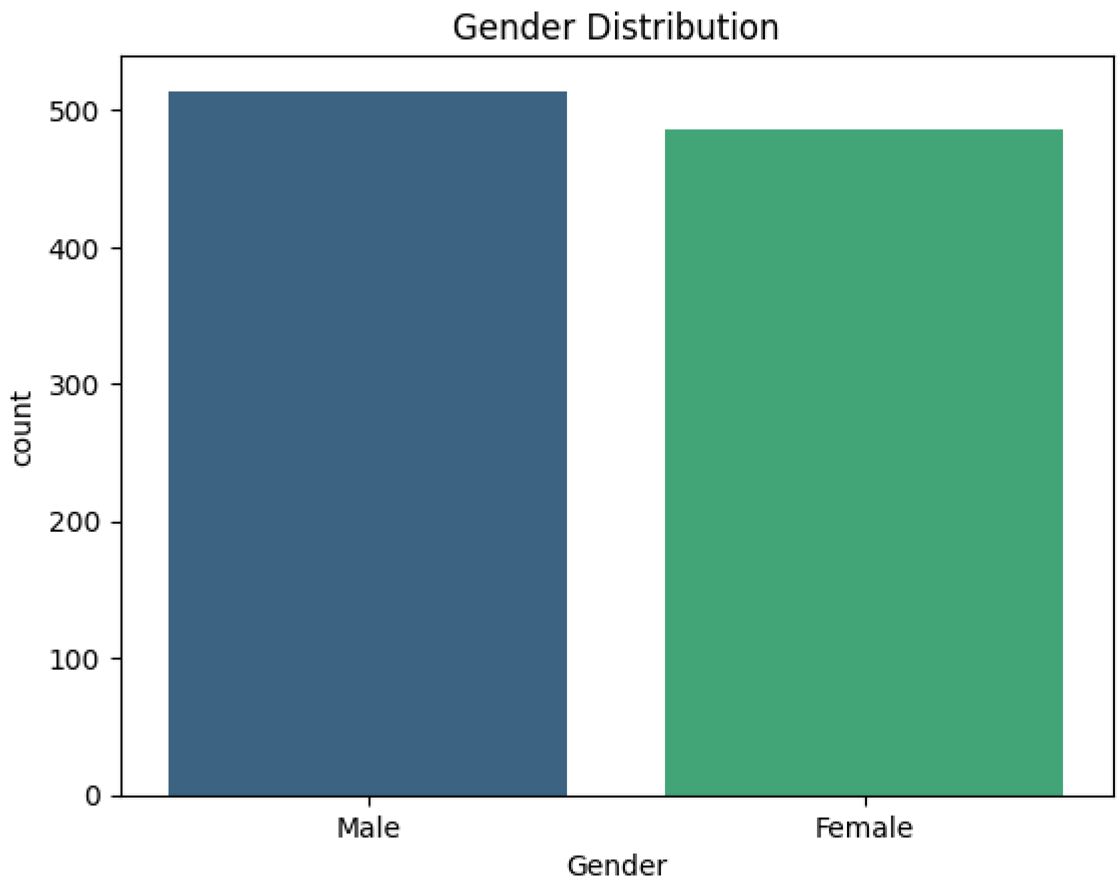
2. Then, **Univariate Analysis** was performed on the datasets to further study the data, the analysis was performed on various columns such as age, BMI, gender, smoking status, etc. Some examples:

```
sns.histplot(df['Age'], kde = True, color = 'blue')
plt.title('Age Distribution')
```

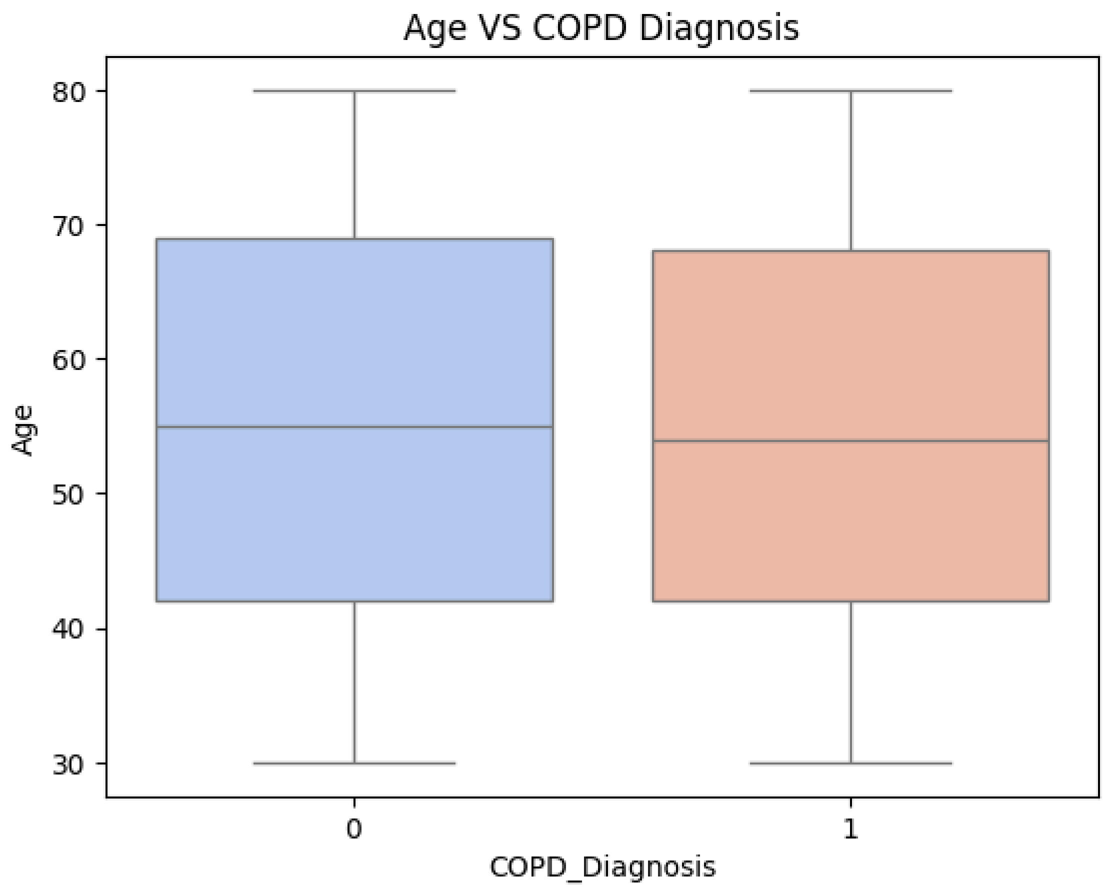
```
ext(0.5, 1.0, 'Age Distribution')
```



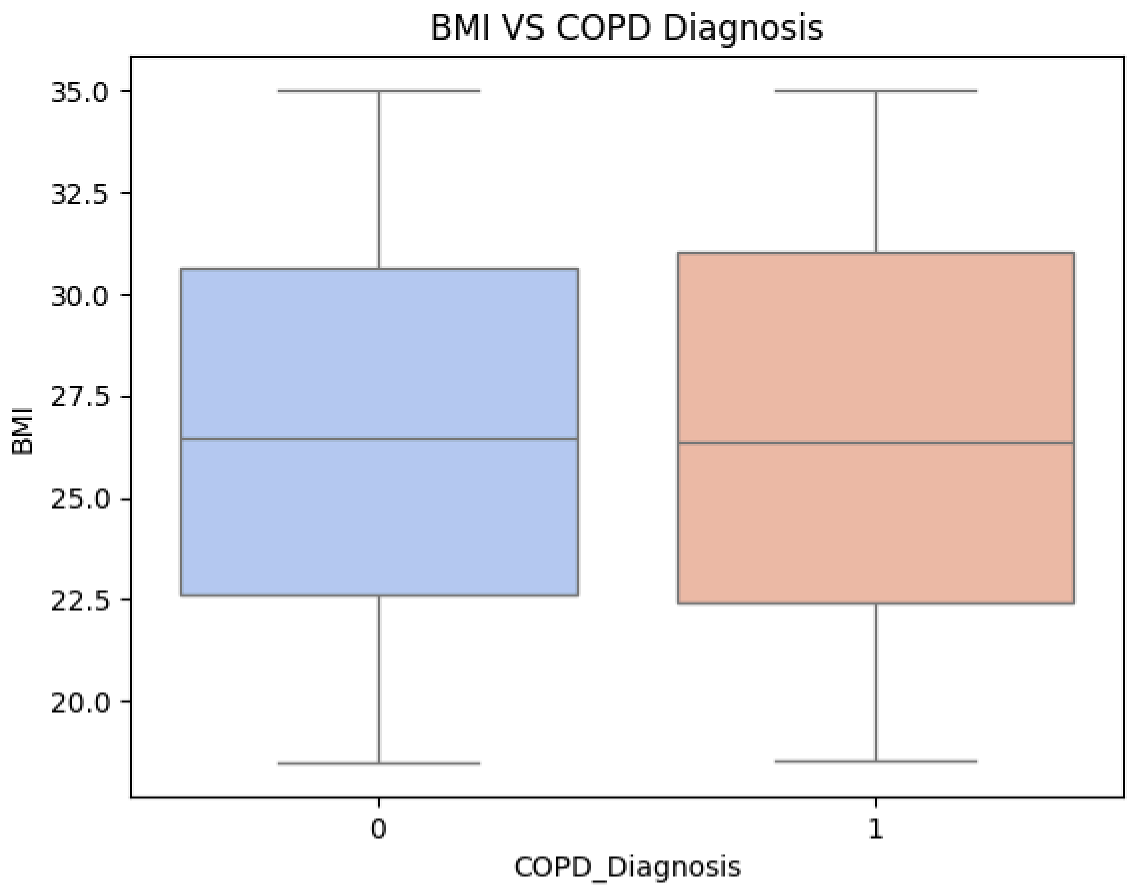

BMI Distribution
Count
0
20
40
60
80
100
20.0
22.5
25.0
27.5
30.0
32.5
35.0
BMI



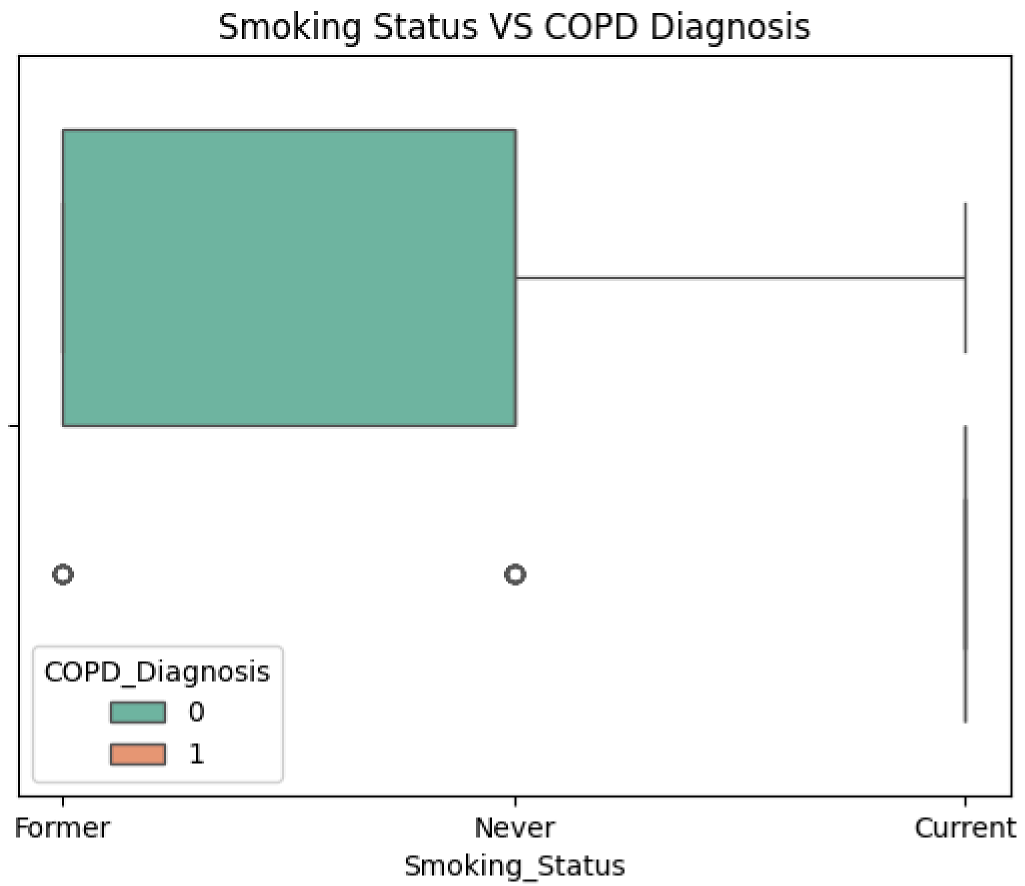
3. After that, **Bivariate Analysis** was done to study the relationship between various features of the datasets, i.e. Age versus COPD Diagnosis, BMI versus COPD Diagnosis, Smoking Status versus COPD Diagnosis, using boxplot and countplot etc. Some examples:

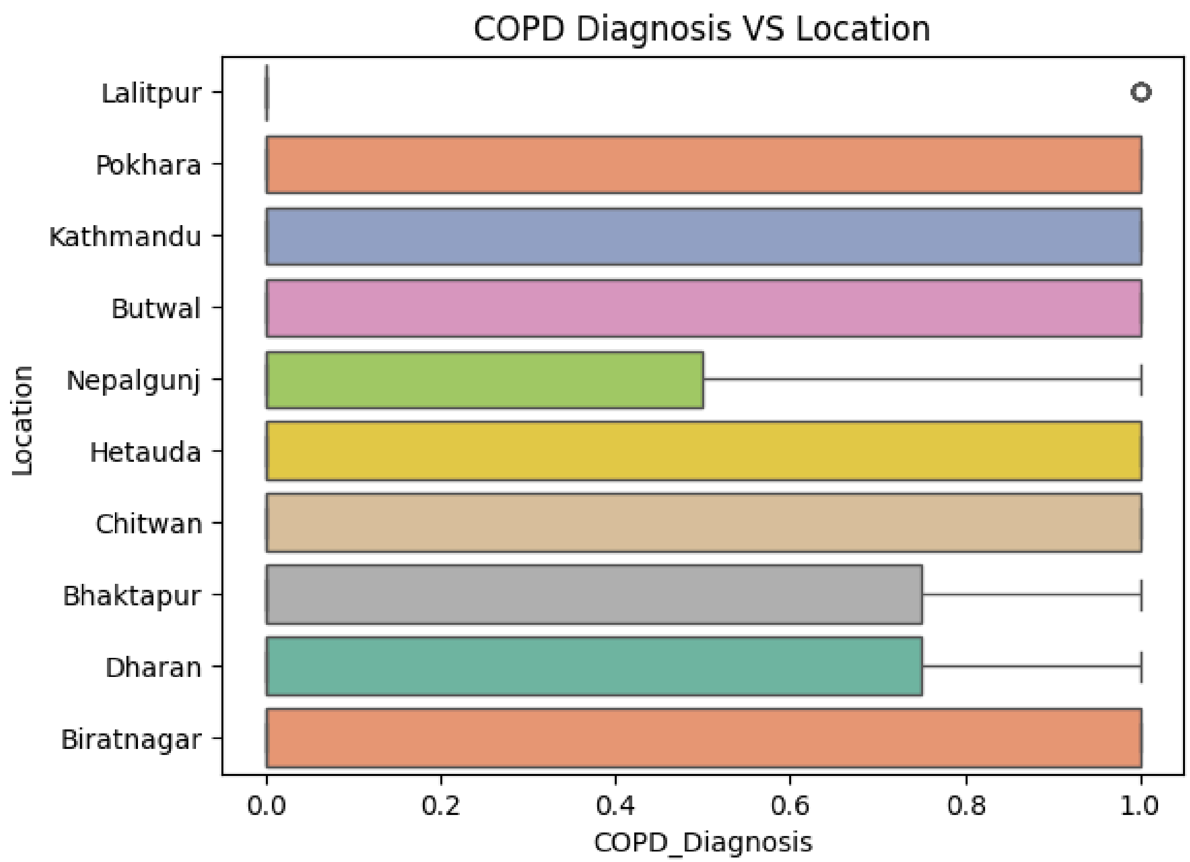


Age VS COPD Diagnosis
80
70
60
50
40
30
Age
0
1
COPD_Diagnosis

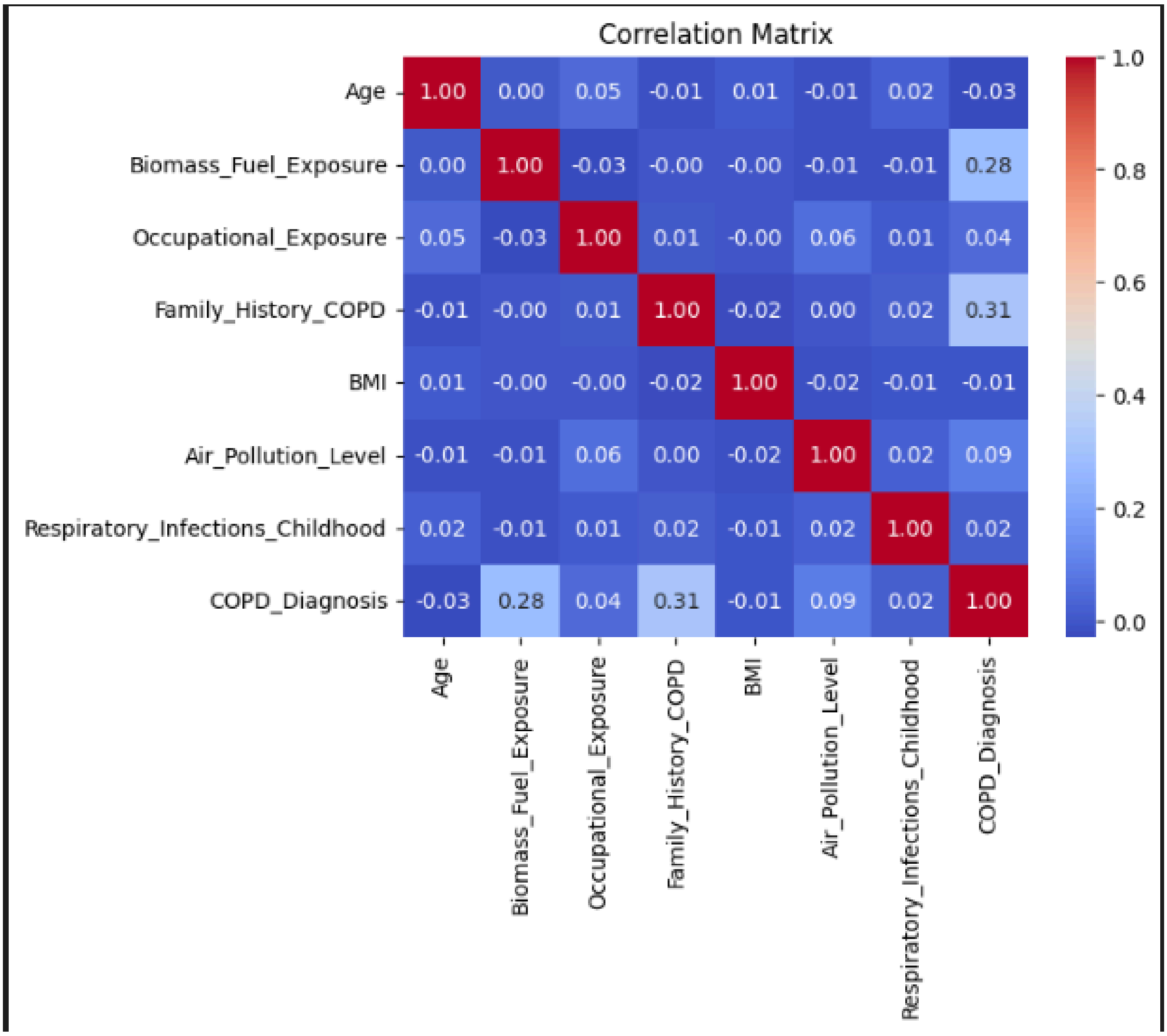


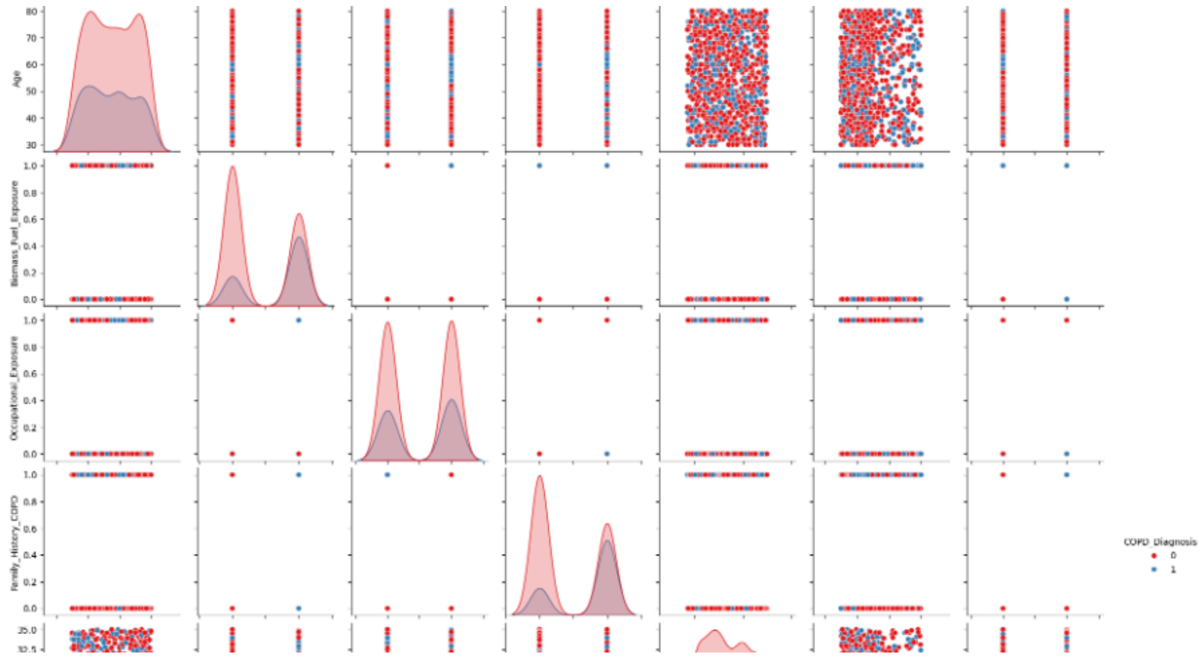
BMI VS COPD Diagnosis
35.0
32.5
30.0
27.5
25.0
22.5
20.0
BMI
0
1
COPD_Diagnosis


Smoking Status VS COPD Diagnosis
COPD_Diagnosis
0
1
Former
Never
Current
Smoking_Status


COPD Diagnosis VS Location
Lalitpur
Pokhara
Kathmandu
Butwal
Nepalgunj
Hetauda
Chitwan
Bhaktapur
Dharan
Biratnagar
Location
0.0
0.2
0.4
0.6
0.8
1.0
COPD_Diagnosis

4. Then, we perform **<u>Multivariate Analysis</u>** to see the relation between the target variable COPD_Diagnosis and rest of the features using correlation matrix and pairplot. Some examples:





Step 5: Feature Engineering

Based on the datasets and research for the domain that was being worked on, various new features were created and some of the existing ones were updated by using ***One-Hot Encoding***.

- Age Category
- BMI Category - Overweight, Normal Weight, Underweight
- Pollution Risk Score - Based on Air Pollution Level
- Smoking Status Encoding
- Interaction Features - Smoking Pollution Interaction based on Smoking Status and Air Pollution Level
- Location Encoding - Categories have to be passed as numbers, changing categories to numerical values

```
 #   Column                             Non-Null Count  Dtype
---  ------                             --------------  -----
 0   Age                                1000 non-null   int64
 1   Gender                             1000 non-null   object
 2   Smoking_Status                     1000 non-null   object
 3   Biomass_Fuel_Exposure              1000 non-null   int64
 4   Occupational_Exposure              1000 non-null   int64
 5   Family_History_COPD                1000 non-null   int64
 6   BMI                                1000 non-null   float64
 7   Air_Pollution_Level                1000 non-null   int64
 8   Respiratory_Infections_Childhood   1000 non-null   int64
 9   COPD_Diagnosis                     1000 non-null   int64
 10  Age_Category                       980 non-null    category
 11  BMI_Category                       1000 non-null   category
 12  Pollution_Risk_Score               1000 non-null   int32
 13  Smoking_Status_Encoded             1000 non-null   float64
 14  Gender_Encoded                     1000 non-null   int64
 15  Smoking_Pollution_Interaction      1000 non-null   float64
 16  Location_Biratnagar                1000 non-null   bool
 17  Location_Butwal                    1000 non-null   bool
 18  Location_Chitwan                   1000 non-null   bool
 19  Location_Dharan                    1000 non-null   bool
...
 23  Location_Nepalgunj                 1000 non-null   bool
 24  Location_Pokhara                   1000 non-null   bool
```

Then, after that, all the features or columns present were studied and then encoded, updated and dropped as necessary since we can only work with integer, float and boolean data-types as shown in the image below:

```
 #   Column                            Non-Null Count  Dtype  
---  ------                            --------------  -----  
 0   Age                               1000 non-null   int64  
 1   Biomass_Fuel_Exposure             1000 non-null   int64  
 2   Occupational_Exposure             1000 non-null   int64  
 3   Family_History_COPD               1000 non-null   int64  
 4   BMI                               1000 non-null   float64
 5   Air_Pollution_Level               1000 non-null   int64  
 6   Respiratory_Infections_Childhood  1000 non-null   int64  
 7   COPD_Diagnosis                    1000 non-null   int64  
 8   Pollution_Risk_Score              1000 non-null   int32  
 9   Smoking_Status_Encoded            1000 non-null   float64
 10  Gender_Encoded                    1000 non-null   int64  
 11  Smoking_Pollution_Interaction     1000 non-null   float64
 12  Location_Biratnagar               1000 non-null   bool   
 13  Location_Butwal                   1000 non-null   bool   
 14  Location_Chitwan                  1000 non-null   bool   
 15  Location_Dharan                   1000 non-null   bool   
 16  Location_Hetauda                  1000 non-null   bool   
 17  Location_Kathmandu                1000 non-null   bool   
 18  Location_Lalitpur                 1000 non-null   bool   
 19  Location_Nepalgunj                1000 non-null   bool   
 20  Location_Pokhara                  1000 non-null   bool   
dtypes: bool(9), float64(3), int32(1), int64(8)
memory usage: 98.8 KB
```

Step 6: Model Development

Then, this updated dataset was split into train and test data sets and then based on the data, COPD Diagnosis(the target variable) is a binary classification which means someone can have COPD(1) or cannot(0), so the following models were used after consideration.

- Logistic Regression
- Decision Trees
- Random Forest

After the datasets were model trained and saved, we evaluated them using accuracy score, precision score and F1 score to figure out the best model as shown below:

```
Logistic Regression Evaluation:
              precision    recall  f1-score   support

           0       0.97      0.98      0.97       134
           1       0.95      0.94      0.95        66

    accuracy                           0.96       200
   macro avg       0.96      0.96      0.96       200
weighted avg       0.96      0.96      0.96       200


Decision Tree Evaluation:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       134
           1       1.00      1.00      1.00        66

    accuracy                           1.00       200
   macro avg       1.00      1.00      1.00       200
weighted avg       1.00      1.00      1.00       200
```

```
Random Forest Evaluation:
              precision    recall  f1-score   support
...
    accuracy                           1.00       200
   macro avg       1.00      1.00      1.00       200
weighted avg       1.00      1.00      1.00       200
```

As we can see both ***"Decision Trees and Random Forest"***, have a **"1.00"** score for accuracy, precision and F1 Score, so both can be nominated for best models. But, even if the score is **"1.00"** there are few things to consider such as ***"Overfitting, Data Imbalance, Test Set Size, etc."*** before concluding it being perfect so any of the two can be chosen for refinement. In this case, **Random Forest** is chosen.

Step 7: Model Tuning and Optimization

For random forest model refinement, we use GridSearchCV to create a refined model along with the best parameters and save it to a pickle file.

```
#Fit the Grid Search CV
grid_search_rf.fit(X_train, y_train)
```

```
GridSearchCV
estimator: RandomForestClassifier
RandomForestClassifier
```

```
#Best Parameters
print(f"Best Parameteres: {grid_search_rf.best_params_}")
best_model = grid_search_rf.best_estimator_
```

```
Best Parameteres: {'max_depth': 10, 'min_samples_split': 2, 'n_estimators': 200}
```

```
#Save the model
with open('Best_Random_Forest_Model.pkl','wb') as f:
    pickle.dump(best_model, f)

print("Model refinement completed and best modelsaved!")
```

```
Model refinement completed and best modelsaved!
```

Step 8: Model Deployment

After this, we used the recently refined and trained datasets to predict COPD. Then deploy the system it to a URL so that anyone will be able to access and use it for COPD prediction.

In the dashboard, various input fields are used to update the features which will predict the COPD and then display the output.

Input Fields:

- Age
- Gender
- BMI
- Smoking Status
- Biomass Fuel Exposure
- Occupational Fuel Exposure
- Family COPD History
- Air Pollution Level
- Respiratory Infections in Childhood
- Location

We use the StreamLit Environment for deployment.

Deployed URL: https://copdprediction-angela.streamlit.app/