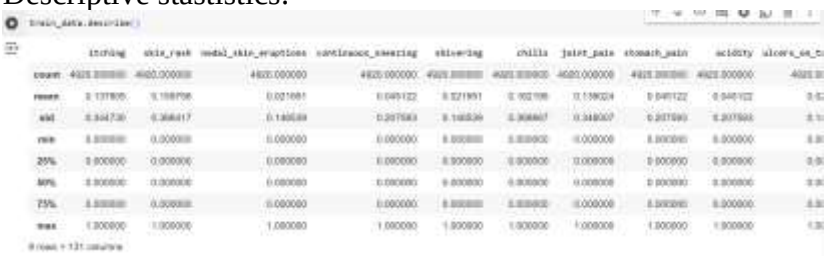
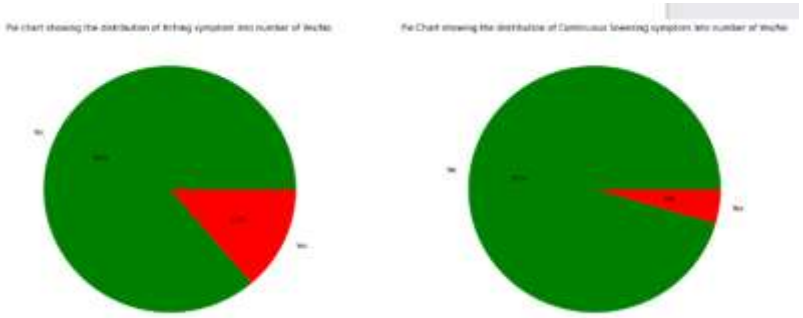
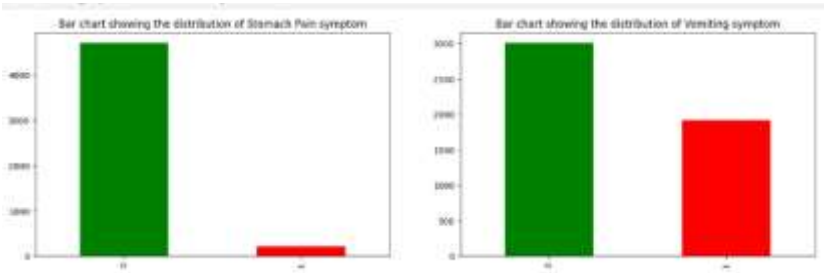
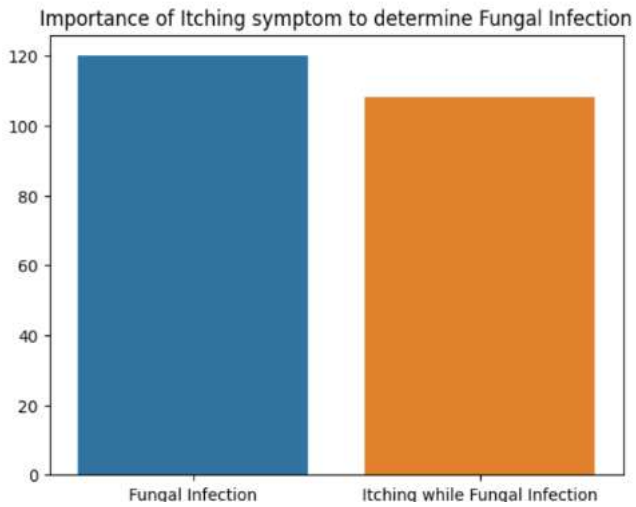
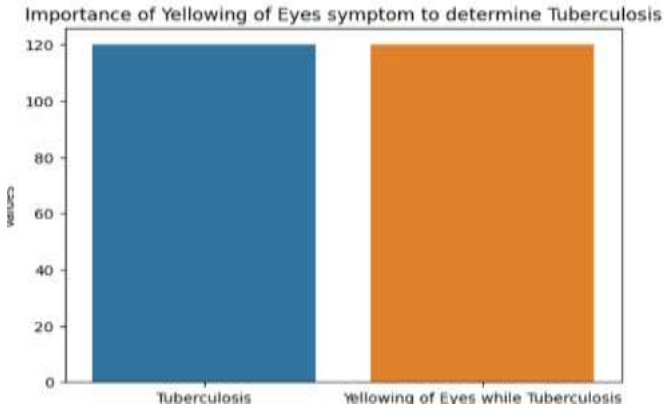
Data Collection and Preprocessing Phase

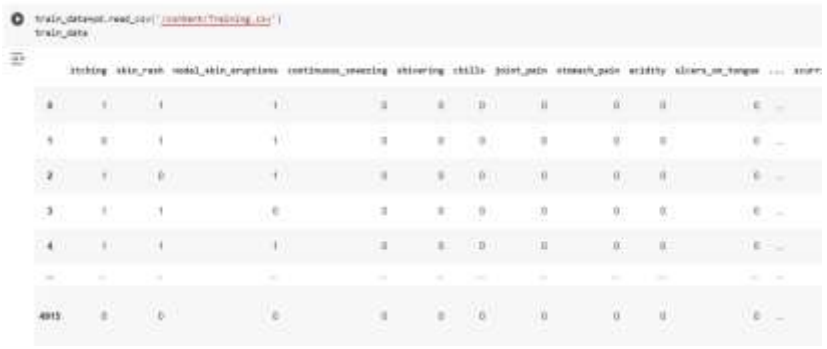
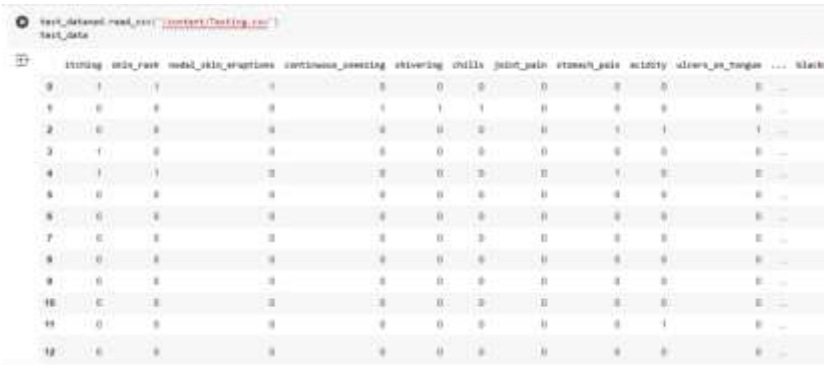
Date	15 June 2024
Team ID	740770
Project Title	Disease prediction using Machine Learning
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template

Dataset variables will be statistically analyzed to identify patterns and outliers, with Python employed for preprocessing tasks like normalization and feature engineering. Data cleaning will address missing values and outliers, ensuring quality for subsequent analysis and modeling, and forming a strong foundation for insights and predictions.

Section	**Description**
Data Overview	Dimension: 8 rows x 131 columns Descriptive stastistics: 
Univariate Analysis	

	
Bivariate Analysis	
	

Multivariate Analysis	
Outliers and Anomalies	-
Data Preprocessing Code Screenshots	
Loading train Data	
Loading test Data	

Handling Missing Data In train and test	

```
[ ] train_data.isnull().sum()
```

```
itching                    0
skin_rash                  0
nodal_skin_eruptions       0
continuous_sneezing        0
shivering                  0
                        ...
blister                    0
red_sore_around_nose       0
yellow_crust_ooze          0
prognosis                  0
Unnamed: 133            4920
Length: 134, dtype: int64
```

```
[ ] train_data.isna().sum().sum()
```

```
4920
```

REMOVING NULL COLUMNS IN TRAINING DATA

```
[ ] train_data['Unnamed: 133'].value_counts()
```

```
Series([], Name: count, dtype: int64)
```

```
[ ] train_data.drop("Unnamed: 133",axis = 1,inplace=True)
    train_data.drop("fluid_overload",axis = 1,inplace=True)
```

```
[ ] train_data.shape
```

```
(4920, 132)
```

```
[ ] test_data.isnull().sum()
```

```
itching                 0
skin_rash               0
nodal_skin_eruptions    0
continuous_sneezing     0
shivering               0
                       ..
inflammatory_nails      0
blister                 0
red_sore_around_nose    0
yellow_crust_ooze       0
prognosis               0
Length: 133, dtype: int64
```

```
test_data.drop("fluid_overload",axis = 1,inplace=True)
```

Data Transformation	
	``python
	from sklearn.preprocessing import LabelEncoder
	label_encoder =LabelEncoder()
	train_data['prognosis']= label_encoder.fit_transform(train_data['prognosis'])
	train_data['prognosis'].unique()
	``
	``
	array([15, 4, 16, 9, 14, 33, 1, 12, 17, 6, 23, 30, 7, 32, 28, 29, 8,
	       11, 37, 40, 19, 20, 21, 22, 3, 36, 10, 34, 13, 18, 39, 26, 24, 25,
	       31, 5, 0, 2, 38, 35, 27])
	``
	``python
	label_encoder =LabelEncoder()
	test_data['prognosis']= label_encoder.fit_transform(test_data['prognosis'])
	test_data['prognosis'].unique()
	``
	``
	array([15, 4, 16, 9, 14, 33, 1, 12, 17, 6, 23, 30, 7, 32, 28, 29, 8,
	       11, 37, 40, 19, 20, 21, 22, 3, 36, 10, 34, 13, 18, 39, 26, 24, 25,
	       31, 5, 0, 2, 38, 35, 27])
	``

Feature Engineering	-
Save Processed Data	-