4. External Search (Information and Data Analysis):

I have gathered additional information from various sources to analyze the data in this content.

1. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8896926/
2. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3661426

So, for this project I am going to use this Dataset:

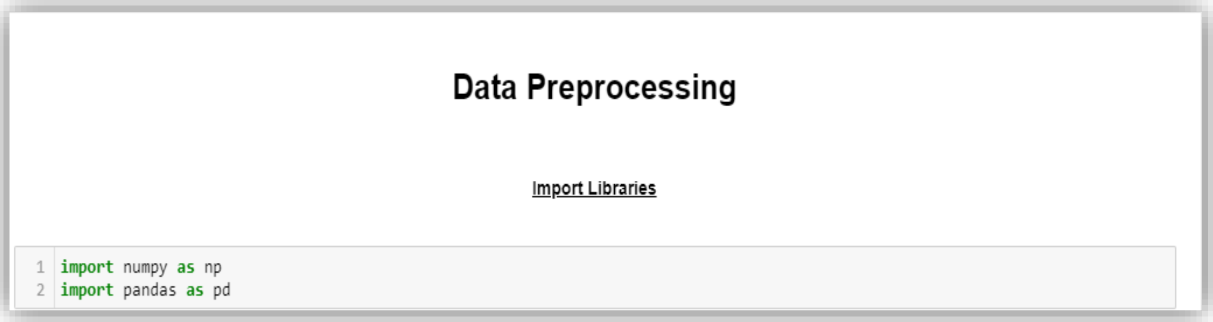
1. Dataset1

Dataset Description:

The first dataset provides information about diseases and their associated symptoms, which is a crucial component of this system. It helps us identify solutions and enables us to inform users about potential problems and their corresponding solutions based on their symptoms.

Now, let's start with the process of this system:

1. Import the libraries for Data Preprocessing:

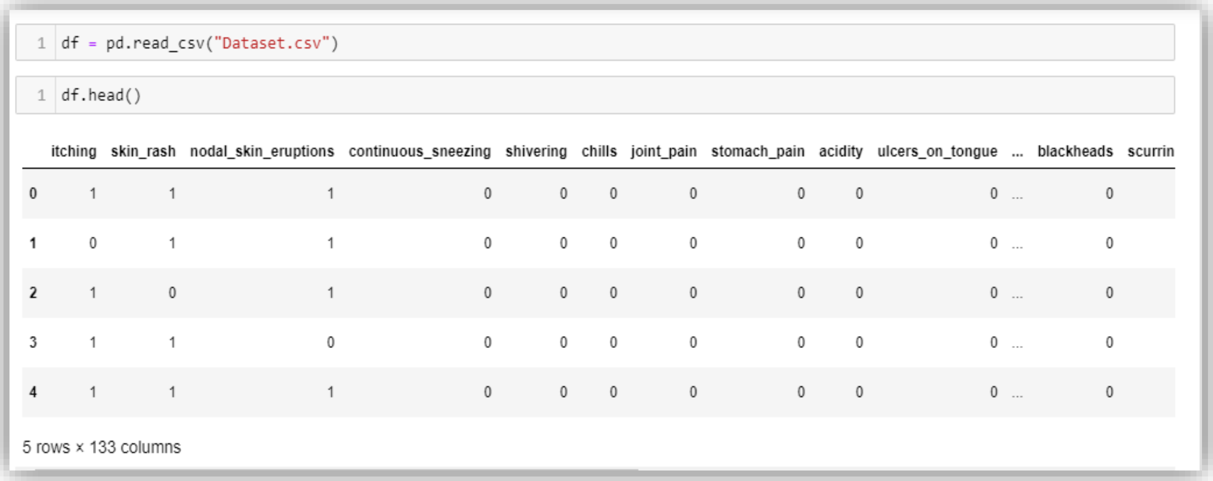


Data Preprocessing

Import Libraries

```
import numpy as np
import pandas as pd
```

Now let's see the Dataset:



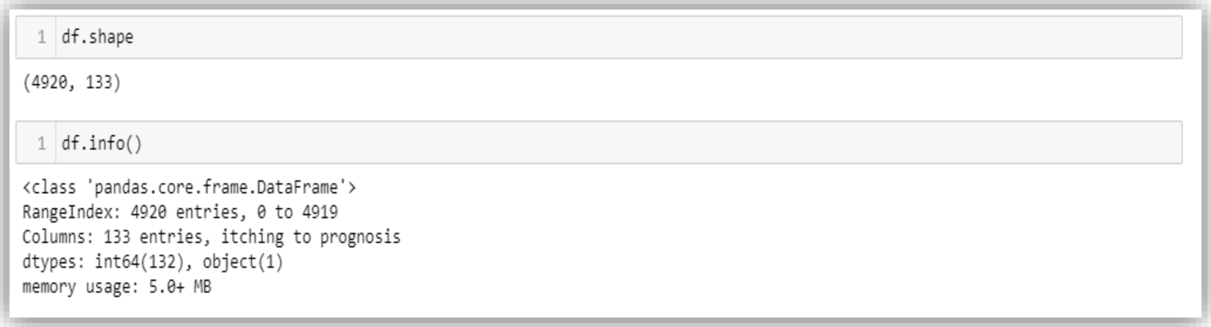
```
df = pd.read_csv("Dataset.csv")
```

```
df.head()
```

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...	blackheads	scurrin
0	1	1	1	0	0	0	0	0	0	0	...	0	
1	0	1	1	0	0	0	0	0	0	0	...	0	
2	1	0	1	0	0	0	0	0	0	0	...	0	
3	1	1	0	0	0	0	0	0	0	0	...	0	
4	1	1	1	0	0	0	0	0	0	0	...	0	

5 rows × 133 columns

Now, it’s time to see the information about dataset:



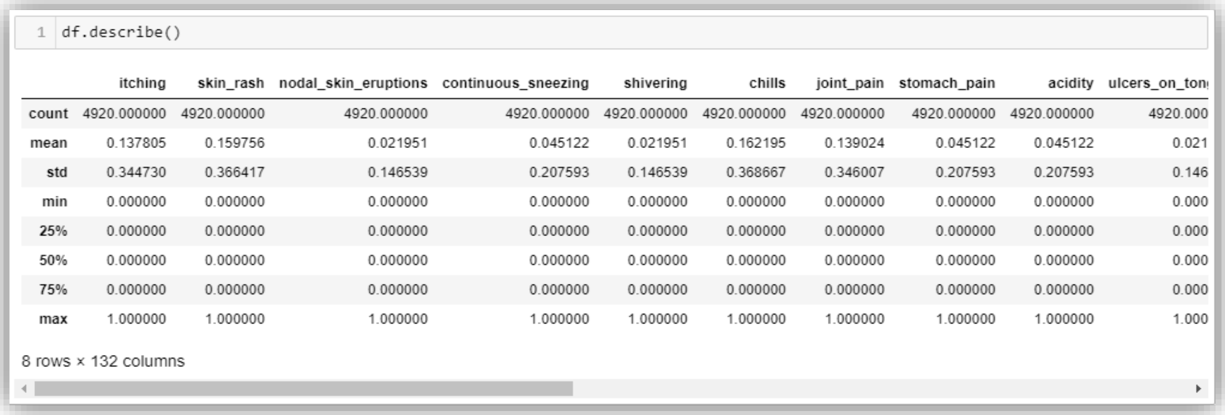
```
df.shape
```

```
(4920, 133)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4920 entries, 0 to 4919
Columns: 133 entries, itching to prognosis
dtypes: int64(132), object(1)
memory usage: 5.0+ MB
```

Description of Dataset:



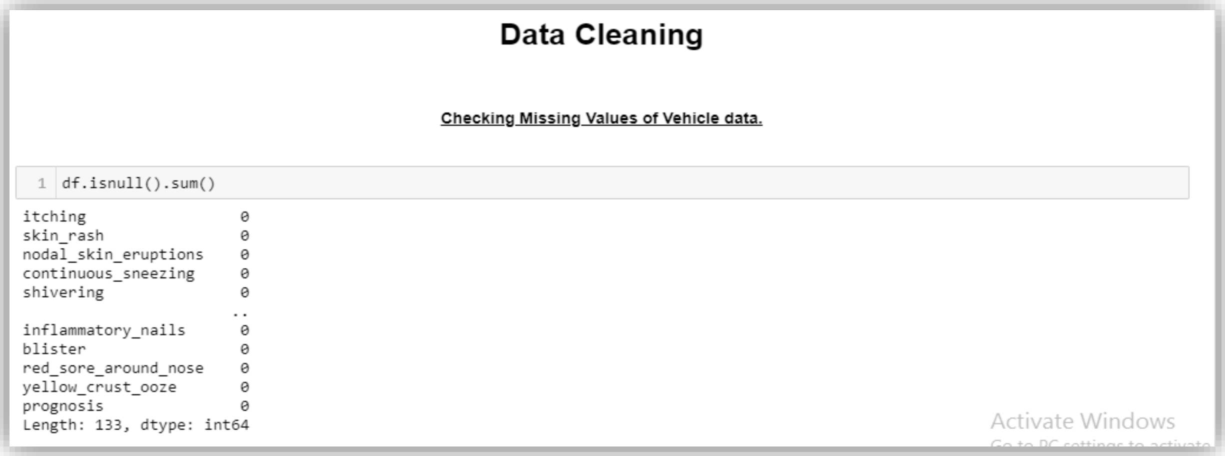
```
df.describe()
```

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_ton
count	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000
mean	0.137805	0.159756	0.021951	0.045122	0.021951	0.162195	0.139024	0.045122	0.045122	0.021
std	0.344730	0.366417	0.146539	0.207593	0.146539	0.368667	0.346007	0.207593	0.207593	0.146
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
75%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000

8 rows × 132 columns

As we can see from the cells above, we have a total of 132 features and one label/class. Depending on the different features, you can determine which disease you are suffering from.

2. It’s time to clean the dataset:



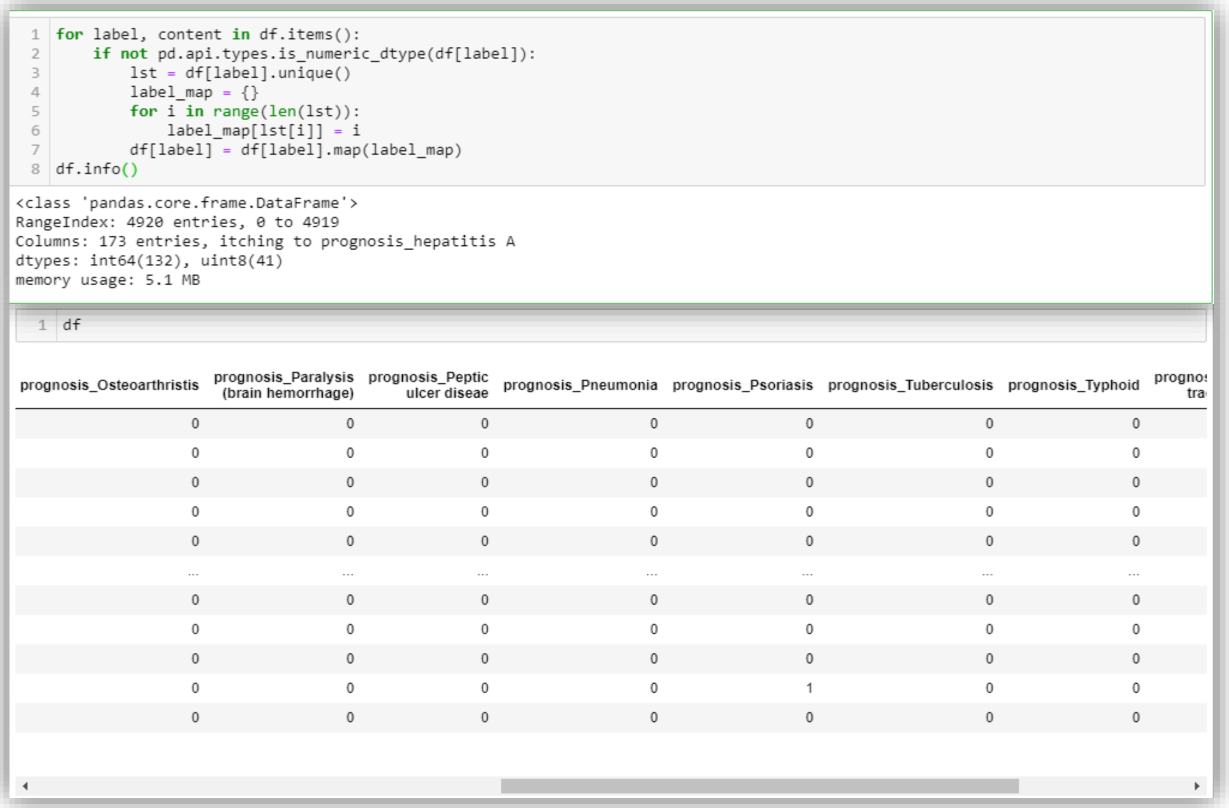
Data Cleaning

Checking Missing Values of Vehicle data.

```
df.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
```

Handling Categorical Data:



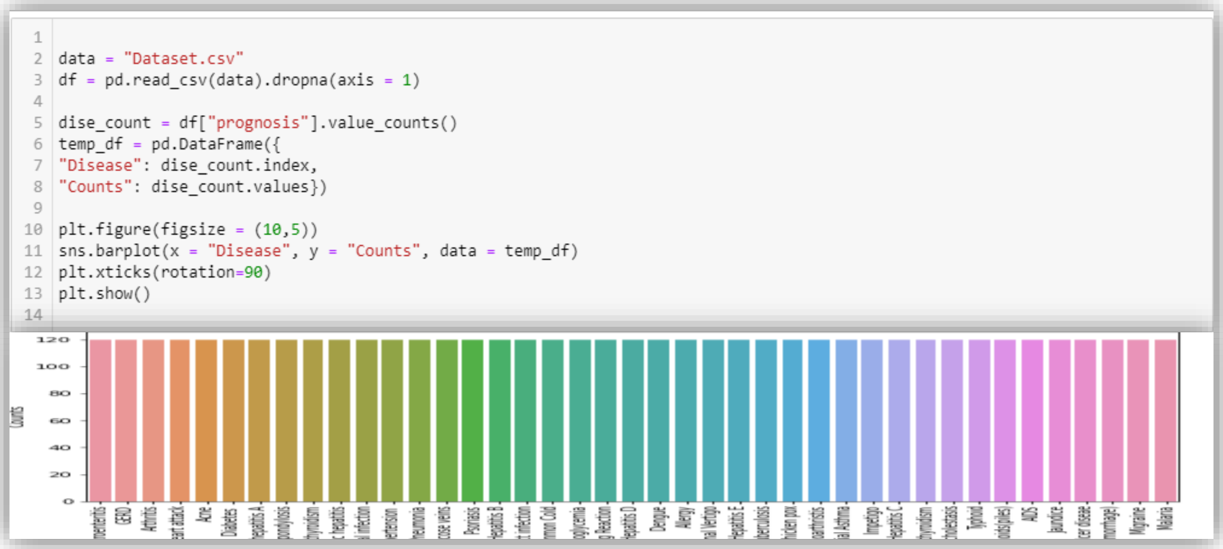
```
for label, content in df.items():
    if not pd.api.types.is_numeric_dtype(df[label]):
        lst = df[label].unique()
        label_map = {}
        for i in range(len(lst)):
            label_map[lst[i]] = i
        df[label] = df[label].map(label_map)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4920 entries, 0 to 4919
Columns: 173 entries, itching to prognosis_hepatitis A
dtypes: int64(132), uint8(41)
memory usage: 5.1 MB
```

```
df
```

prognosis_Osteoarthristis	prognosis_Paralysis (brain hemorrhage)	prognosis_Peptic ulcer diseae	prognosis_Pneumonia	prognosis_Psoriasis	prognosis_Tuberculosis	prognosis_Typhoid	prognos tra
0	0	0	0	0	0	0	
0	0	0	0	0	0	0	
0	0	0	0	0	0	0	
0	0	0	0	0	0	0	
0	0	0	0	0	0	0	
...	...	...	...	...	...	...	
0	0	0	0	0	0	0	
0	0	0	0	0	0	0	
0	0	0	0	0	0	0	
0	0	0	0	1	0	0	
0	0	0	0	0	0	0	

To check whether the dataset is balanced or not by using Bar Plot:



```

data = "Dataset.csv"
df = pd.read_csv(data).dropna(axis = 1)

dise_count = df["prognosis"].value_counts()
temp_df = pd.DataFrame({
"Disease": dise_count.index,
"Counts": dise_count.values})

plt.figure(figsize = (10,5))
sns.barplot(x = "Disease", y = "Counts", data = temp_df)
plt.xticks(rotation=90)
plt.show()

```

As we can see, the dataset is completely balanced, and we have already converted our categorical class into numerical ones for a better view.

5. Benchmarking for Disease Prediction System:

- **Introduction:**
  Working on the Performance and Capabilities of this system, benchmarking played a very crucial role. An overview of the benchmarking process, goals, methodology, and its effects on our project are given in this section.

- **Objectives of Benchmarking:**
  Our benchmarking guided by the following objectives:
  - To evaluate the “Disease prediction system” based on the speed, accuracy and overall performance of this system.
  - Compare our system with industrial standards and their solutions.
  - Identify the area for improvement of our system’s algorithm and user interface.

- **Methodology:**
  The benchmarking process involved the following key steps:
  - **Data Collection:**
    We analyze various datasets, data sources to understand the specific requirements of our system, ensuring it covered various disease and symptoms.

  - **Selection of Benchmarks:**
    We found that, industry standards and benchmark systems that are well-known for their superior disease prediction capabilities.

- **Analysis and Comparison:**
  Based on the benchmarking data, we observed the following:
  - We recognize the need of continuous improvement in our search of excellence.
  - We aim to generate better results and outputs because we are always looking for better ways to do things.
  - This is a commitment to the health and welfare of our users as well as to the success of our project, as they depend on our system for accurate illness forecasts, medical advice, and easy access to a large database of medical data.

- **Conclusion of Benchmarking:**
  The practice of benchmarking played a crucial role in evaluating and improving this system .We were able to improve system performance and bring it into line with the standards of the industry as a result.

6. Applicable Patents:

- **https://patents.google.com/patent/CN109036553B/en?q=(Disease+Prediction+system)&oq=Disease+Prediction+system**
- **https://patents.google.com/patent/KR101884609B1/en?q=(Disease+Prediction+system)&oq=Disease+Prediction+system**
- **https://patents.google.com/patent/EP2652684B1/en?q=(pateint+Diagnosis+Prediction+system)&oq=pateint+Diagnosis+Prediction+system**

7. Applicable Regulations (Government and Environmental):

- Healthcare Data Regulations.
- Data Privacy and Protection.
- Ethical Considerations.