**Problem Statement**

To build a Multi-classification methodology to determine whether a person facing which diseases and what required to do.

Our intuitive interface allows users to input their symptoms effortlessly, creating a seamless user experience.

**Architecture:**



**Data Description**

The patient will send data in multiple sets of files in batches at a given location. The data has been extracted from the csv files.

The data contains 131 rows and 1 target column.

**Features:**

1. itching
2. skin\_rash
3. nodal\_skin\_eruptions
4. continuous\_sneezing
5. shivering
6. chills
7. joint\_pain
8. stomach\_pain
9. acidity
10. ulcers\_on\_tongue
11. muscle\_wasting
12. vomiting
13. burning\_micturition
14. spotting\_urination
15. fatigue
16. weight\_gain
17. anxiety
18. cold\_hands\_and\_feets
19. mood\_swings
20. weight\_loss
21. restlessness
22. lethargy
23. patches\_in\_throat
24. irregular\_sugar\_level
25. cough
26. high\_fever
27. sunken\_eyes
28. breathlessness
29. sweating
30. dehydration
31. indigestion
32. headache
33. yellowish\_skin
34. dark\_urine
35. nausea
36. loss\_of\_appetite
37. pain\_behind\_the\_eyes
38. back\_pain
39. constipation
40. abdominal\_pain
41. diarrhoea
42. mild\_fever
43. yellow\_urine
44. yellowing\_of\_eyes
45. acute\_liver\_failure
46. fluid\_overload
47. swelling\_of\_stomach
48. swelled\_lymph\_nodes
49. malaise
50. blurred\_and\_distorted\_vision
51. phlegm
52. throat\_irritation
53. redness\_of\_eyes
54. sinus\_pressure
55. runny\_nose
56. congestion
57. chest\_pain
58. weakness\_in\_limbs
59. fast\_heart\_rate
60. pain\_during\_bowel\_movements
61. pain\_in\_anal\_region
62. bloody\_stool
63. irritation\_in\_anus
64. neck\_pain
65. dizziness
66. cramps
67. bruising
68. obesity
69. swollen\_legs
70. swollen\_blood\_vessels
71. puffy\_face\_and\_eyes
72. enlarged\_thyroid
73. brittle\_nails
74. swollen\_extremeties
75. excessive\_hunger
76. extra\_marital\_contacts
77. drying\_and\_tingling\_lips
78. slurred\_speech
79. knee\_pain
80. hip\_joint\_pain
81. muscle\_weakness
82. stiff\_neck
83. swelling\_joints
84. movement\_stiffness
85. spinning\_movements
86. loss\_of\_balance
87. unsteadiness
88. weakness\_of\_one\_body\_side
89. loss\_of\_smell
90. bladder\_discomfort
91. foul\_smell\_of urine
92. continuous\_feel\_of\_urine
93. passage\_of\_gases
94. internal\_itching
95. toxic\_look\_(typhos)
96. depression
97. irritability
98. muscle\_pain
99. altered\_sensorium
100. red\_spots\_over\_body
101. belly\_pain
102. abnormal\_menstruation
103. dischromic \_patches
104. watering\_from\_eyes
105. increased\_appetite
106. polyuria
107. family\_history
108. mucoid\_sputum
109. rusty\_sputum
110. lack\_of\_concentration
111. visual\_disturbances
112. receiving\_blood\_transfusion
113. receiving\_unsterile\_injections
114. coma': 113, 'stomach\_bleeding
115. distention\_of\_abdomen
116. history\_of\_alcohol\_consumption
117. fluid\_overload.1
118. blood\_in\_sputum
119. prominent\_veins\_on\_calf
120. palpitations
121. painful\_walking
122. pus\_filled\_pimples
123. blackheads
124. scurring
125. skin\_peeling
126. silver\_like\_dusting
127. small\_dents\_in\_nails
128. inflammatory\_nails
129. blister
130. red\_sore\_around\_nose
131. yellow\_crust\_ooze

**Target Label:**

**Prognosis:**

Fungal infection

Allergy

GERD

Chronic cholestasis

Drug Reaction

Peptic ulcer diseae

AIDS

Diabetes

Gastroenteritis

Bronchial Asthma

Hypertension

Migraine

Cervical spondylosis

Paralysis (brain hemorrhage)

Jaundice

Malaria

Chicken pox

Dengue

Typhoid

hepatitis A

Hepatitis B

Hepatitis C

Hepatitis D

Hepatitis E

Alcoholic hepatitis

Tuberculosis

Common Cold

Pneumonia

Dimorphic hemmorhoids(piles)

Heart attack

Varicose veins

Hypothyroidism

Hyperthyroidism

Hypoglycemia

Osteoarthristis

Arthritis

(vertigo) Paroymsal Positional Vertigo

Acne

Urinary tract infection

Psoriasis

Impetigo

**Data Validation**

In this step, we perform different sets of validation on the given set of training files.

1. Name Validation- We validate the name of the files based on the given name in the schema file. We have created a regex pattern as per the name given in the schema file to use for validation. After validating the pattern in the name, we check for the length of date in the file name as well as the length of time in the file name. If all the values are as per requirement, we move such files to "Good\_Data\_Folder" else we move such files to "Bad\_Data\_Folder."
2. Number of Columns - We validate the number of columns present in the files, and if it doesn't match with the value given in the schema file, then the file is moved to "Bad\_Data\_Folder."
3. Name of Columns - The name of the columns is validated and should be the same as given in the schema file. If not, then the file is moved to "Bad\_Data\_Folder".
4. The datatype of columns - The datatype of columns is given in the schema file. It is validated when we insert the files into Database. If the datatype is wrong, then the file is moved to "Bad\_Data\_Folder".
5. Null values in columns - If any of the columns in a file have all the values as NULL or missing, we discard such a file and move it to "Bad\_Data\_Folder".

**Data Insertion in Database**

1) Database Creation and connection - Create a database with the given name passed. If the database has already been created, open a connection to the database.

2) Table creation in the database - Table with name - "Good\_Data", is created in the database for inserting the files in the "Good\_Data\_Folder" based on given column names and datatype in the schema file. If the table is already present, then the new table is not created, and new files are inserted in the already present table as we want training to be done on new as well as old training files.

3) Insertion of files in the table - All the files in the "Good\_Data\_Folder" are inserted in the above-created table. If any file has invalid data type in any of the columns, the file is not loaded in the table and is moved to "Bad\_Data\_Folder".

**Model Training**

1) **Data Export from Db** - The data in a stored database is exported as a CSV file to be used for model training.

2) **Data Preprocessing**

1. Check for null values in the columns. If present, impute the null values using the categorical imputer.
2. Scale the numeric values using the standard scaler.
3. Check for correlation.

3) **Clustering -** KMeans algorithm is used to create clusters in the preprocessed data. The optimum number of clusters is selected by plotting the elbow plot, and for the dynamic selection of the number of clusters, we are using "KneeLocator" function. The idea behind clustering is to implement different algorithms

The Kmeans model is trained over preprocessed data, and the model is saved for further use in prediction.

1. **Model Selection –** After the clusters have been created, we find the best model for each cluster. We are using algorithms like:

RandomForestClassifier

GradientBoostingClassifier

KNeighborsClassifier

MultinomialNB.

For each cluster, these algorithms are passed with the best parameters derived from GridSearch. We calculate the accuracy scores for all models and select the model with the best score. Similarly, the model is selected for each cluster. All the models for every cluster are saved for use in prediction.

**Prediction Data Description**

The patient will send the input in multiple sets of information at a given location.

**Data Validation**

In this step, we perform different sets of validation on the given set of training files.

1) Name Validation- We validate the name of the files based on given Name in the schema file. We have created a regex pattern as per the name given in the schema file, to use for validation. After validating the pattern in the name, we check for the length of date in the file name as well as the length of the timestamp in the file name. If all the values are as per requirement, we move such files to "Good\_Data\_Folder" else we move such files to "Bad\_Data\_Folder".

2) Number of Columns - We validate the number of columns present in the files, and if it doesn't match with the value given in the schema file, then the file is moved to "Bad\_Data\_Folder".

3) Name of Columns - The name of the columns is validated and should be same as given in the schema file. If not, then the file is moved to "Bad\_Data\_Folder".

4) Datatype of columns - The datatype of columns is given in the schema file. This is validated when we insert the files into Database. If the datatype is incorrect, then the file is moved to "Bad\_Data\_Folder".

5) Null values in columns - If any of the columns in a file has all the values as NULL or missing, we discard such file and move it to "Bad\_Data\_Folder".

**Data Insertion in Database**

1) Database Creation and connection - Create a database with the given name passed. If the database is already created, open the connection to the database.

2) Table creation in the database - Table with name - "Good\_Data", is created in the database for inserting the files in the "Good\_Data\_Folder" based on given column names and datatype in the schema file. If the table is already present, then a new table is not created, and new files are inserted into the already present table as we want training to be done on new as well old training files.

3) Insertion of files in the table - All the files in the "Good\_Data\_Folder" are inserted in the above-created table. If any file has invalid data type in any of the columns, the file is not loaded in the table and is moved to "Bad\_Data\_Folder".

**Prediction**

1) Data Export from Db - The data in the stored database is exported as a CSV file to be used for prediction.

2) Data Preprocessing :

1. Check for null values in the columns. If present, impute the null values using the categorical imputer.
2. Scale the numeric values using the standard scaler.
3. Check for correlation.

3) Clustering - KMeans model created during training is loaded, and clusters for the preprocessed prediction data is predicted.

4) Prediction - Based on the cluster number, the respective model is loaded and is used to predict the data for that cluster.

5) Once the prediction is made for all the clusters, the predictions along with the Wafer names are saved in a CSV file at a given location, and the location is returned to the client.