Chatbot Risk Monitoring in Healthcare

Problem Formulation/Explanation

The fast life that we are living now makes us forgot about our health and ignore our health lifestyle with everyday stress a possibility to have a chronic disease is increasing another situation is having a small disease symptom that has been ignored till these symptoms turn to be a series disease and can cause death in the worst scenario. So, we proposed a healthcare chatbot that triggers the health status of a patient and develop a healthcare chatbot that lets the users understand the symptoms they are facing and get a basic diagnosis about the diseases they could be having.

Virtual assistants who communicate via text are helping to manage drugs, monitor chronic health issues, and recognize symptoms. The use of smartphones, along with the growing popularity of health applications, IoT, telehealth, and other related technologies, is boosting the healthcare market. Virtual assistants, for example, save time and lighten the workload for doctors. The main advantage of employing chatbots is that customers can ask any question without being aware of the right keywords. Chatbots can easily understand natural language by comparing the words associated with the question, and they can then offer accurate answers quickly and easily.

Methodology

- **Collect Dataset**
- **Data Preprocessing**
- **Exploratory Data analysis**
- **Input Text preprocessing**
- **Syntactic Similarity**
- **Semantic Similarity**
- **Classification and Clustering**
- **Association rule**

Collect Dataset

Raw dataset

Disease with symptoms

41 diseases, 132 symptoms

	A	B	C	D	E	F
1	Disease	Symptom_1	Symptom_2	Symptom_3	Symptom_4	Symptom_5
2	Fungal infection	itching	skin_rash	nodal_skin_eruptions	dischromic _patches	
3	Fungal infection	skin_rash	nodal_skin_eruptions	dischromic _patches		
4	Fungal infection	itching	nodal_skin_eruptions	dischromic _patches		
5	Fungal infection	itching	skin_rash	dischromic _patches		
6	Fungal infection	itching	skin_rash	nodal_skin_eruptions		
7	Fungal infection	skin_rash	nodal_skin_eruptions	dischromic _patches		
8	Fungal infection	itching	nodal_skin_eruptions	dischromic _patches		
9	Fungal infection	itching	skin_rash	dischromic _patches		
10	Fungal infection	itching	skin_rash	nodal_skin_eruptions		
11	Fungal infection	itching	skin_rash	nodal_skin_eruptions	dischromic _patches	
12	Allergy	continuous_sneezing	shivering	chills	watering_from_eyes	
13	Allergy	shivering	chills	watering_from_eyes		
14	Allergy	continuous_sneezing	chills	watering_from_eyes		
15	Allergy	continuous_sneezing	shivering	watering_from_eyes		
16	Allergy	continuous_sneezing	shivering	chills		
17	Allergy	shivering	chills	watering_from_eyes		
18	Allergy	continuous_sneezing	chills	watering_from_eyes		
19	Allergy	continuous_sneezing	shivering	watering_from_eyes		
20	Allergy	continuous_sneezing	shivering	chills		
21	Allergy	continuous_sneezing	shivering	chills	watering_from_eyes	

Symptoms' severity

	A	B
1	Symptom	weight
2	itching	1
3	skin_rash	3
4	nodal_skin_eruptions	4
5	continuous_sneezing	4
6	shivering	5
7	chills	3
8	joint_pain	3
9	stomach_pain	5
10	acidity	3
11	ulcers_on_tongue	4
12	muscle_wasting	3
13	vomiting	5
14	burning_micturition	6
15	spotting_urination	6
16	fatigue	4
17	weight_gain	3
18	anxiety	4
19	cold_hands_and_feets	5
20	mood_swings	3

Disease description

	A	B
1	Disease	Description
2	Drug Reaction	An adverse drug reaction (ADR) is an injury caused by taking medication. ADRs may occur following a single dose or prolonged administration of a drug or result from the combination of two or more drugs.
3	Malaria	An infectious disease caused by protozoan parasites from the Plasmodium family that can be transmitted by the bite of the Anopheles mosquito or by a contaminated needle or transfusion. Falciparum malaria is the
4	Allergy	An allergy is an immune system response to a foreign substance that's not typically harmful to your body.They can include certain foods, pollen, or pet dander. Your immune system's job is to keep you healthy by fight
5	Hypothyroidism	Hypothyroidism, also called underactive thyroid or low thyroid, is a disorder of the endocrine system in which the thyroid gland does not produce enough thyroid hormone.
6	Psoriasis	Psoriasis is a common skin disorder that forms thick, red, bumpy patches covered with silvery scales. They can pop up anywhere, but most appear on the scalp, elbows, knees, and lower back. Psoriasis can't be passed
7	GERD	Gastroesophageal reflux disease, or GERD, is a digestive disorder that affects the lower esophageal sphincter (LES), the ring of muscle between the esophagus and stomach. Many people, including pregnant women, s
8	Chronic cholestasis	Chronic cholestatic diseases, whether occurring in infancy, childhood or adulthood, are characterized by defective bile acid transport from the liver to the intestine, which is caused by primary damage to the biliary ep
9	hepatitis A	Hepatitis A is a highly contagious liver infection caused by the hepatitis A virus. The virus is one of several types of hepatitis viruses that cause inflammation and affect your liver's ability to function.
10	Osteoarthristis	Osteoarthritis is the most common form of arthritis, affecting millions of people worldwide. It occurs when the protective cartilage that cushions the ends of your bones wears down over time.
11	(vertigo) Paroymsal Position	Benign paroxysmal positional vertigo (BPPV) is one of the most common causes of vertigo â€" the sudden sensation that you're spinning or that the inside of your head is spinning. Benign paroxysmal positional vertigo
12	Hypoglycemia	Hypoglycemia is a condition in which your blood sugar (glucose) level is lower than normal. Glucose is your body's main energy source. Hypoglycemia is often related to diabetes treatment. But other drugs and a varie
13	Acne	Acne vulgaris is the formation of comedones, papules, pustules, nodules, and/or cysts as a result of obstruction and inflammation of pilosebaceous units (hair follicles and their accompanying sebaceous gland). Acne c
14	Diabetes	Diabetes is a disease that occurs when your blood glucose, also called blood sugar, is too high. Blood glucose is your main source of energy and comes from the food you eat. Insulin, a hormone made by the pancreas
15	Impetigo	Impetigo (im-puh-TIE-go) is a common and highly contagious skin infection that mainly affects infants and children. Impetigo usually appears as red sores on the face, especially around a child's nose and mouth, and o
16	Hypertension	Hypertension (HTN or HT), also known as high blood pressure (HBP), is a long-term medical condition in which the blood pressure in the arteries is persistently elevated. High blood pressure typically does not cause syr
17	Peptic ulcer diseae	Peptic ulcer disease (PUD) is a break in the inner lining of the stomach, the first part of the small intestine, or sometimes the lower esophagus. An ulcer in the stomach is called a gastric ulcer, while one in the first part

Disease precaution

	A	B	C	D	E
1	Disease	Precaution_1	Precaution_2	Precaution_3	Precaution_4
2	Drug Reaction	stop irritation	consult nearest hospital	stop taking drug	follow up
3	Malaria	Consult nearest hospital	avoid oily food	avoid non veg food	keep mosquitos out
4	Allergy	apply calamine	cover area with bandage		use ice to compress itching
5	Hypothyroidism	reduce stress	exercise	eat healthy	get proper sleep
6	Psoriasis	wash hands with warm soapy water	stop bleeding using pressure	consult doctor	salt baths
7	GERD	avoid fatty spicy food	avoid lying down after eating	maintain healthy weight	exercise
8	Chronic cholestasis	cold baths	anti itch medicine	consult doctor	eat healthy
9	hepatitis A	Consult nearest hospital	wash hands through	avoid fatty spicy food	medication
10	Osteoarthristis	acetaminophen	consult nearest hospital	follow up	salt baths
11	(vertigo) Paroymsal Positional Vertigo	lie down	avoid sudden change in body	avoid abrupt head movment	relax
12	Hypoglycemia	lie down on side	check in pulse	drink sugary drinks	consult doctor
13	Acne	bath twice	avoid fatty spicy food	drink plenty of water	avoid too many products
14	Diabetes	have balanced diet	exercise	consult doctor	follow up
15	Impetigo	soak affected area in warm water	use antibiotics	remove scabs with wet compressed cloth	consult doctor
16	Hypertension	meditation	salt baths	reduce stress	get proper sleep
17	Peptic ulcer diseae	avoid fatty spicy food	consume probiotic food	eliminate milk	limit alcohol

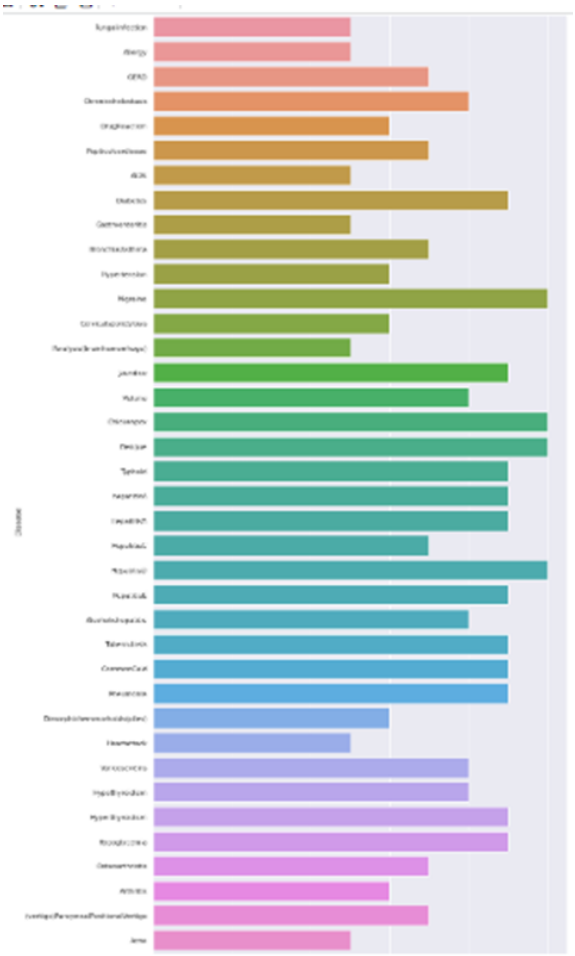
Data Preprocessing

1. Remove whitespaces
2. Remove duplicates (len_before= 4920, len_after=304)
3. Set each row to a unique disease and each column with unique symptom and fill the cells with the symptoms weight.

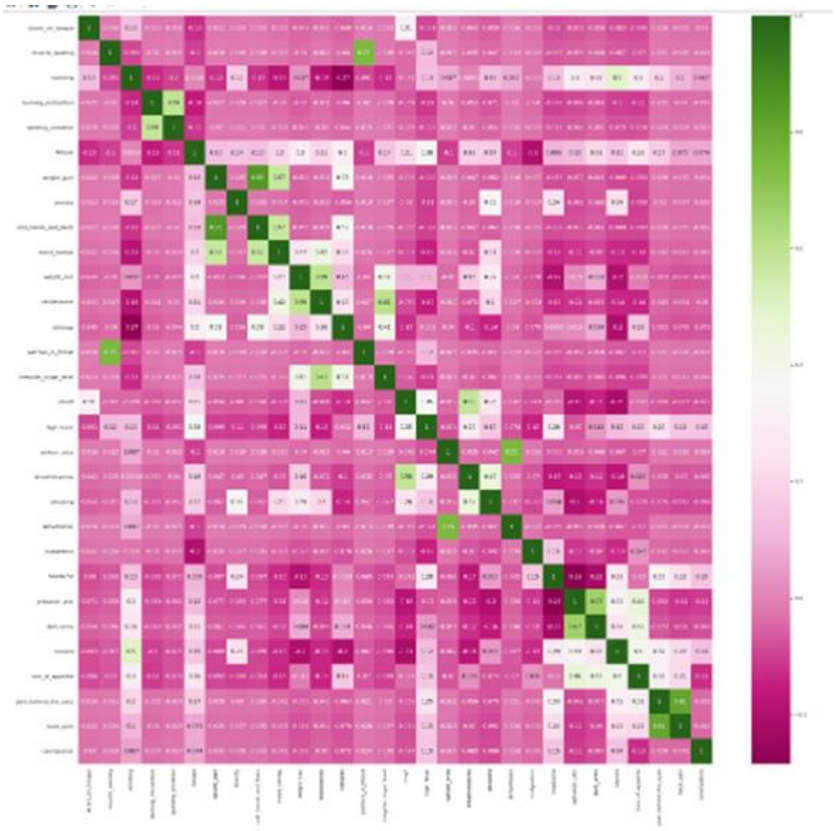
Disease	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	muscle_wasting	vomiting	burning_micturition	spotting_urination
Fungalinfection	1	3	4	0	0	0	0	0	0	0	0	0	0	0
Fungalinfection	0	3	4	0	0	0	0	0	0	0	0	0	0	0
Fungalinfection	1	0	4	0	0	0	0	0	0	0	0	0	0	0
Fungalinfection	1	3	0	0	0	0	0	0	0	0	0	0	0	0
Fungalinfection	1	3	4	0	0	0	0	0	0	0	0	0	0	0
Allergy	0	0	0	4	5	3	0	0	0	0	0	0	0	0
Allergy	0	0	0	0	5	3	0	0	0	0	0	0	0	0
Allergy	0	0	0	4	0	3	0	0	0	0	0	0	0	0
Allergy	0	0	0	4	5	0	0	0	0	0	0	0	0	0
Allergy	0	0	0	4	5	3	0	0	0	0	0	0	0	0
GERD	0	0	0	0	0	0	0	5	3	4	0	5	0	0
GERD	0	0	0	0	0	0	0	5	0	4	0	5	0	0
GERD	0	0	0	0	0	0	0	5	3	0	0	5	0	0

Exploratory Data analysis

Visualization of existence diseases



linear relationships between some of features using correlation heatmap: which symptoms occur together?



Feature Engineering

Text Input preprocessing

Objective is to extract the symptom from an input text and map it to the symptoms of the dataset the input sentence walks through a pipeline

1- **Preprocess the symptoms of the dataset** remove all the underscores, stop words, and lemmatize the word.

```
[39] all_symp_pr=[preprocess(sym) for sym in all_symp]

     all_symp_pr

     ['itch',
      'skin rash',
      'nodal skin eruption',
      'continuous sneeze',
      'shiver',
      'chill',
      'joint pain',
      'stomach pain',
      'acidity',
      'ulcer tongue',
```

2- Associates each preprocessed symptoms with the name of its original column

```
col_dict = dict(zip(all_symp_pr, all_symp_col))
```

```
col_dict
```

```
{'abdominal pain': 'abdominal_pain',
 'abnormal menstruation': 'abnormal_menstruation',
 'acidity': 'acidity',
 'acute liver failure': 'acute_liver_failure',
 'altered sensorium': 'altered_sensorium',
 'anxiety': 'anxiety',
 'belly pain': 'belly_pain',
```

3- **Preprocess the input sentence:** Remove stop words and lemmatize the word to its source.

```
preprocess_sym('my skin has some nodal eruptions')
```

```
'skin nodal eruption'
```

4- Calculate the syntactic similarity between preprocessed input sentences and all preprocessed symptoms using Jaccard similarity.
The syntactic similarity assumes that the similarity between the two texts is proportional to the number of identical words in them.

```
syntactic_similarity(preprocess_sym('my skin has some nodal eruptions') ,all_symp_pr)
```

```
(1, ['nodal skin eruption'])
```

5- In case there wasn't any syntactic similarity we calculate semantic similarity between preprocessed input sentences and all preprocessed symptoms.
Semantic similarity focuses more on the meaning and interpretation-based similarity between the two texts.

```
suggest_syn('puke')
```

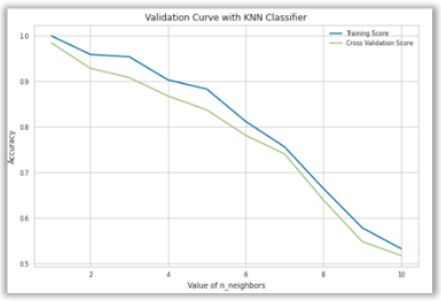
```
['vomit', 'pain', 'swollen blood vessel', 'dark urine']
```

Classification and Clustering

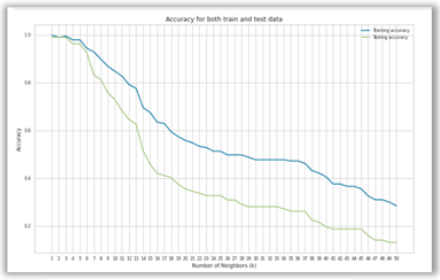
1. **Supervised Machine Learning**

We applied supervised learning to the data for diagnosis ang get the disease using classification techniques such as KNN, Decision tree, Support vector machine, Random forest, these classifiers can predict what the disease the patient inquiry. we plot the cross validation to get the bias and variance for each model.

- **KNN classifier**



Cross validation for KNN model



train and test accuracy

The accuracy of **KNN model** is **96.2%** with **K=5**

- **Random Forest classifier**

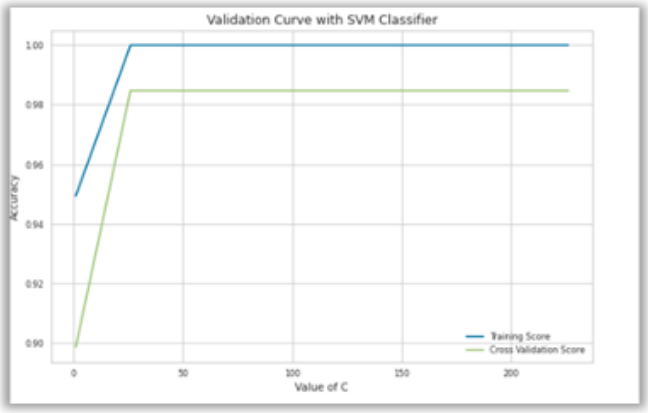
```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(random_state = 42, n_estimators = 20)
rfc.fit(X_train, y_train.values.ravel())
rfc.predict(X_test)
rfc.score(X_test, y_test)
```

The accuracy of Random Forest model is **100 %**

- **Support Vector Machine**

```
from sklearn.svm import SVC
svc = SVC(gamma = "auto", kernel = "rbf" )
svc.fit(X_train, y_train.values.ravel())
svc.predict(X_test)
svc.score(X_test, y_test)
print(svc.score(X_test, y_test))
param_range = np.arange(1, 250, 25)

plot_cv_indices(svc, X_train, y_train,"C", param_range , cv=2,model_name = "SVM")
```



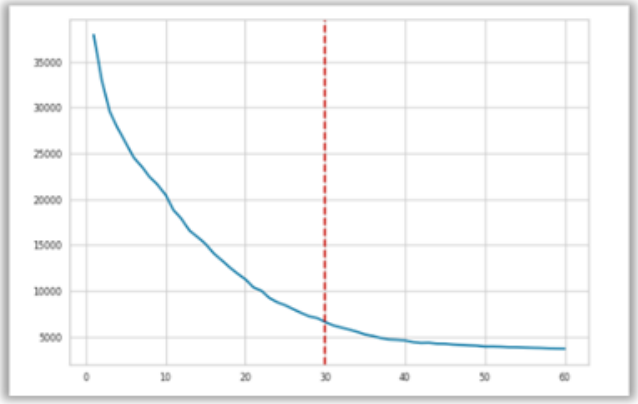
train andtest accuracy

The accuracy of SVM model is **94.3%**

2. **Unsupervised Machine Learning**

We applied unsupervised learning to the data for diagnosis ang get the disease using clustering techniques such as K-Means this model can predict what the disease the patient inquiry. we plot WSS to get the number of clusters for model.

```python
wcss = []
for number_of_clusters in range(1, 61):
    kmeans = KMeans(n_clusters = number_of_clusters, random_state = 42)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
ks = range(1, 61)
plt.plot(ks, wcss)
plt.axvline(30, linestyle='--', color='r')
```



WSS plot

Evaluation of ML Results

We made evaluation matrix for each model and get report classification also confusion matrix to compare between them and choose the champion model based on accuracy.

- **KNN**

```python
y_predictions = {"KNN": knn.predict(X_test),
         "SVC": svc.predict(X_test),
         "DT": dt.predict(X_test),
         "RFC": rfc.predict(X_test)}

from sklearn.metrics import classification_report, confusion_matrix, ConfusionMatrixDisplay

for classifier, y_pred in y_predictions.items():
    cm = confusion_matrix(y_test, y_pred)
    print(classifier,'Confusion matrix: \n',cm)
    print("------------------")
    print(classifier, 'Classification report: \n',classification_report(y_test,y_pred, labels=np.unique(y_pred)))
```

- **Confusion matrix for each classifier:**

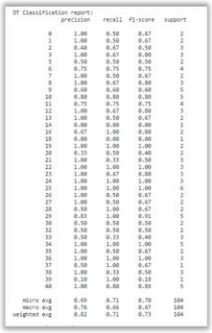
```
KNN Confusion matrix:
 [[2 0 0 ... 0 0 0]
 [0 2 0 ... 0 0 0]
 [0 0 3 ... 0 0 0]
 ...
 [0 0 1 ... 0 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 5]]
------------------
```

```
SVC Confusion matrix:
 [[2 0 0 ... 0 0 0]
 [0 2 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 3 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 5]]
------------------
```

```
DT Confusion matrix:
 [[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 1 0]
 [0 0 2 ... 0 1 0]
 ...
 [0 0 0 ... 1 1 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 4]]
------------------
```

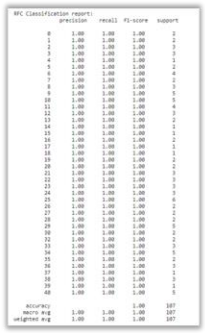
```
RFC Confusion matrix:
 [[2 0 0 ... 0 0 0]
 [0 2 0 ... 0 0 0]
 [0 0 3 ... 0 0 0]
 ...
 [0 0 0 ... 3 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 5]]
------------------
```

- **Report classification for each classifier:**



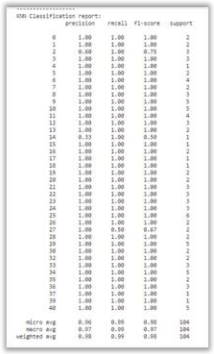
DT Classification report:

	precision	recall	f1-score	support
0	1.00	0.50	0.67	2
1	1.00	0.50	0.67	2
2	0.40	0.67	0.50	3
3	1.00	0.67	0.80	3
5	0.50	0.50	0.50	2
6	0.75	0.75	0.75	4
7	1.00	0.50	0.67	2
8	1.00	0.67	0.80	3
9	0.60	0.60	0.60	5
10	0.80	0.80	0.80	5
11	0.75	0.75	0.75	4
12	1.00	0.67	0.80	3
13	1.00	0.50	0.67	2
14	0.00	0.00	0.00	1
16	0.67	1.00	0.80	2
18	0.00	0.00	0.00	1
19	1.00	1.00	1.00	2
20	0.33	0.50	0.40	2
21	1.00	0.33	0.50	3
22	1.00	1.00	1.00	3
23	1.00	0.67	0.80	3
24	1.00	1.00	1.00	3
25	1.00	1.00	1.00	6
26	1.00	0.50	0.67	2
27	1.00	0.50	0.67	2
28	0.50	1.00	0.67	2
29	0.83	1.00	0.91	5
30	0.50	0.50	0.50	2
32	0.50	0.50	0.50	2
33	0.50	0.33	0.40	3
34	1.00	1.00	1.00	5
35	1.00	0.50	0.67	2
36	1.00	1.00	1.00	3
37	0.50	1.00	0.67	1
38	1.00	0.33	0.50	3
39	0.10	1.00	0.18	1
40	1.00	0.80	0.89	5
micro avg	0.65	0.71	0.70	104
macro avg	0.76	0.66	0.67	104
weighted avg	0.82	0.71	0.73	104



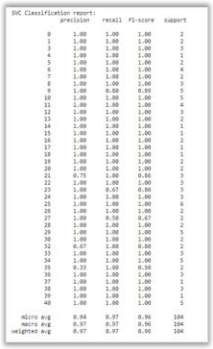
RFC Classification report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2
1	1.00	1.00	1.00	2
2	1.00	1.00	1.00	3
3	1.00	1.00	1.00	3
4	1.00	1.00	1.00	1
5	1.00	1.00	1.00	2
6	1.00	1.00	1.00	4
7	1.00	1.00	1.00	2
8	1.00	1.00	1.00	3
9	1.00	1.00	1.00	5
10	1.00	1.00	1.00	5
11	1.00	1.00	1.00	4
12	1.00	1.00	1.00	3
13	1.00	1.00	1.00	2
14	1.00	1.00	1.00	1
15	1.00	1.00	1.00	1
16	1.00	1.00	1.00	2
17	1.00	1.00	1.00	1
18	1.00	1.00	1.00	1
19	1.00	1.00	1.00	2
20	1.00	1.00	1.00	2
21	1.00	1.00	1.00	3
22	1.00	1.00	1.00	3
23	1.00	1.00	1.00	3
24	1.00	1.00	1.00	3
25	1.00	1.00	1.00	6
26	1.00	1.00	1.00	2
27	1.00	1.00	1.00	2
28	1.00	1.00	1.00	2
29	1.00	1.00	1.00	5
30	1.00	1.00	1.00	2
32	1.00	1.00	1.00	2
33	1.00	1.00	1.00	3
34	1.00	1.00	1.00	5
35	1.00	1.00	1.00	2
36	1.00	1.00	1.00	3
37	1.00	1.00	1.00	1
38	1.00	1.00	1.00	3
39	1.00	1.00	1.00	1
40	1.00	1.00	1.00	5
accuracy			1.00	107
macro avg	1.00	1.00	1.00	107
weighted avg	1.00	1.00	1.00	107



KNN Classification report:

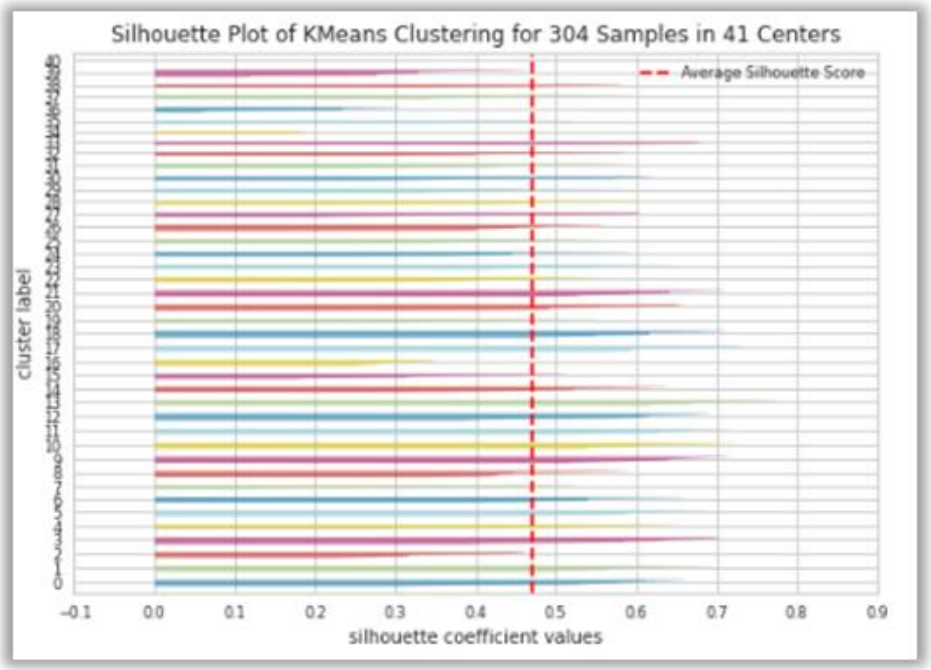
	precision	recall	f1-score	support
0	1.00	1.00	1.00	2
1	1.00	1.00	1.00	2
2	0.60	1.00	0.75	3
3	1.00	1.00	1.00	3
4	1.00	1.00	1.00	1
5	1.00	1.00	1.00	2
6	1.00	1.00	1.00	4
7	1.00	1.00	1.00	2
8	1.00	1.00	1.00	3
9	1.00	1.00	1.00	5
10	1.00	1.00	1.00	5
11	1.00	1.00	1.00	4
12	1.00	1.00	1.00	3
13	1.00	1.00	1.00	2
14	0.33	1.00	0.50	1
15	1.00	1.00	1.00	1
16	1.00	1.00	1.00	2
17	1.00	1.00	1.00	1
18	1.00	1.00	1.00	1
19	1.00	1.00	1.00	2
20	1.00	1.00	1.00	2
21	1.00	1.00	1.00	3
22	1.00	1.00	1.00	3
23	1.00	1.00	1.00	3
24	1.00	1.00	1.00	3
25	1.00	1.00	1.00	6
26	1.00	1.00	1.00	2
27	1.00	0.50	0.67	2
28	1.00	1.00	1.00	2
29	1.00	1.00	1.00	5
30	1.00	1.00	1.00	2
32	1.00	1.00	1.00	2
33	1.00	1.00	1.00	3
34	1.00	1.00	1.00	5
35	1.00	1.00	1.00	2
36	1.00	1.00	1.00	3
37	1.00	1.00	1.00	1
38	1.00	1.00	1.00	3
39	0.10	1.00	0.18	1
40	1.00	0.80	0.89	5
micro avg	0.86	0.99	0.90	104
macro avg	0.97	0.99	0.97	104
weighted avg	0.98	0.99	0.98	104



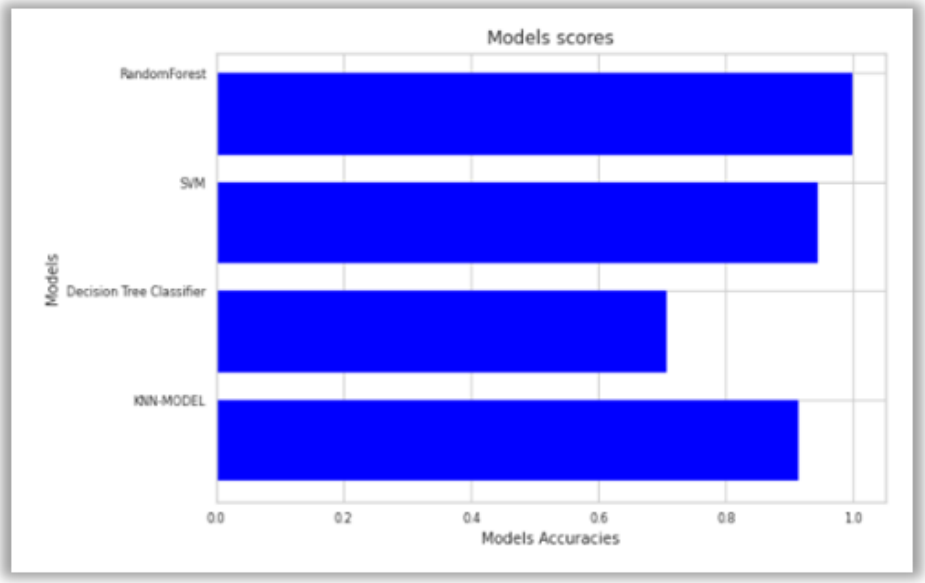
SVC Classification report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2
1	1.00	1.00	1.00	2
3	1.00	1.00	1.00	3
4	1.00	1.00	1.00	1
5	1.00	1.00	1.00	2
6	1.00	1.00	1.00	4
7	1.00	1.00	1.00	2
8	1.00	1.00	1.00	3
9	1.00	0.80	0.89	5
10	1.00	1.00	1.00	5
11	1.00	1.00	1.00	4
12	1.00	1.00	1.00	3
13	1.00	1.00	1.00	2
14	1.00	1.00	1.00	1
15	1.00	1.00	1.00	1
16	1.00	1.00	1.00	2
17	1.00	1.00	1.00	1
18	1.00	1.00	1.00	1
19	1.00	1.00	1.00	2
20	1.00	1.00	1.00	2
21	0.75	1.00	0.86	3
22	1.00	1.00	1.00	3
23	1.00	0.67	0.80	3
24	1.00	1.00	1.00	3
25	1.00	1.00	1.00	6
26	1.00	1.00	1.00	2
27	1.00	0.50	0.67	2
28	1.00	1.00	1.00	2
29	1.00	1.00	1.00	5
30	1.00	1.00	1.00	2
32	0.67	1.00	0.80	2
33	1.00	1.00	1.00	3
34	1.00	1.00	1.00	5
35	0.33	1.00	0.50	2
36	1.00	1.00	1.00	3
37	1.00	1.00	1.00	1
38	1.00	1.00	1.00	3
39	1.00	1.00	1.00	1
40	1.00	1.00	1.00	5
micro avg	0.94	0.97	0.96	104
macro avg	0.97	0.97	0.96	104
weighted avg	0.97	0.97	0.96	104

- **Silhouette score for K-Means**



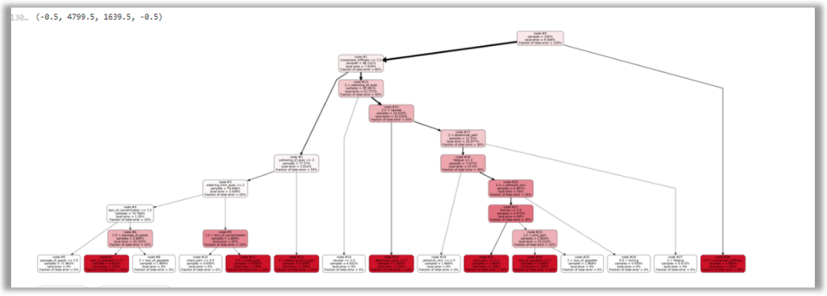
- **Models Accuracies**

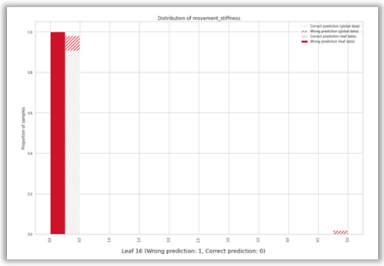


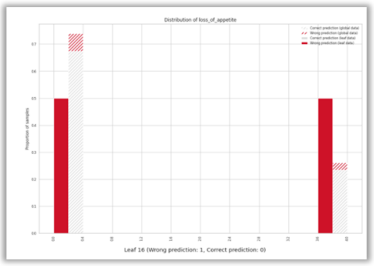
Random forest is the best model with highest accuracy

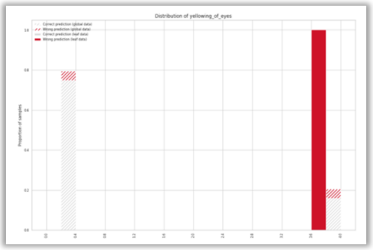
Error Analysis

From modeling and analysis, we select the champion model with highest accuracy is Random forest.









Integration

Chatbots are software tools developed to interact with people through chat. The first chatbots could construct straightforward dialogues based on a complicated set of principles. Using Python and Dialog flow frameworks, we tried to show our result through a chatbot that can be more simple, user-friendly, and flexible for users. In order to diagnose and analyze disease and display the results of the diagnostic through a series of questions between the user and the chatbot, we created a chatbot.

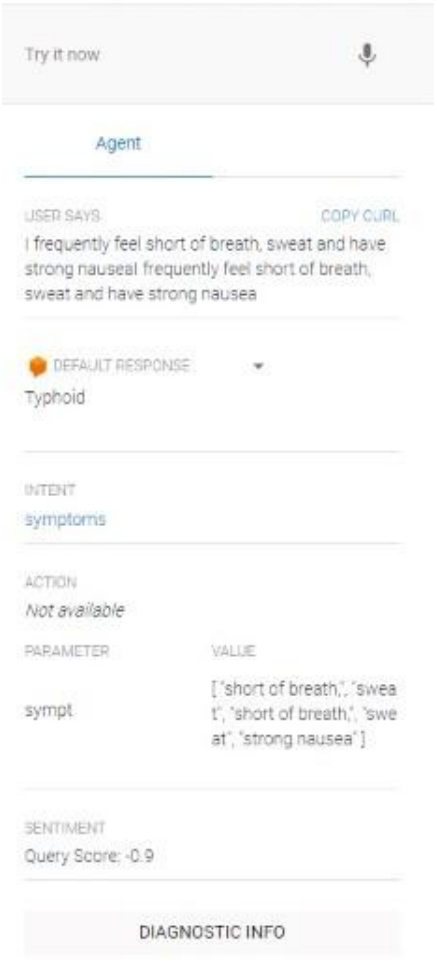
We integrate our python code that can diagnose the symptoms of a patient and retrieve the disease through many machine learning algorithms to predict or classify the diseases based on symptoms.

Steps to Add Dialog flow Chatbot to Python Frameworks:

1. **Create an agent**
   First, we log in to Dialog flow and create the agent with training phrases and corresponding responses to handle expected conversation scenarios with the end-user. We created DR. Bot Agent.
2. **Create an Intent**
   After creating the DR. Bot agent, we added an intent called symptoms for communication with the user so that we could understand the expressions that they were writing.
3. **Enable Fulfillment**

After adding symptoms intent, we did not add agent manual responses in the Responses section. Since we were using Ngrok for the same, we enabled webhook for this intent. The webhook will help us transfer data and responses between Dialog flow and python code. Dialog flow provides webhook services via Dialog flow Fulfillment.
In the fulfillment section, we enabled the Webhook option and paste the URL generated in Ngrok, and append /webhook at the end

In the figures below, we enabled the connection to get and transfer the responses.
After the connection was established, we tried the integration
using the input **“I frequently feel short of breath, sweat, and have strong nausea”**
The response was **Typhoid disease**



Innovativeness

What is your contribution?

As sometimes patients don't know the disease they have, we introduced a feature to diagnose the disease instead of just a question-answering system. We developed a bot that can kindly ask about a patient's health and give a basic diagnosis even before a doctor's visit. The bot can diagnose the disease based on the symptoms that the patient provided.

What sets your project apart from others?
In our project, we implemented a chatbot to diagnose and predict the disease based on symptoms from the end-user or the patient. First, we used a dataset that contains diseases and symptoms for each disease, we picked up a dataset that has 143 diseases with their symptoms as a Database to retrieve the result. We applied pre-processing and feature engineering to the dataset to extract the basic symptoms from the complaints of the patient to map the symptoms to the disease in the dataset.
One of the related works to our project just employed supervised learning techniques such as SVM and Decision tree.
In our work, we applied many supervised and unsupervised techniques:
Supervised:

- Classification (KNN, DT, SVM, Random Forest)

Unsupervised:

- Clustering (K means)
- Association rules

We implemented Association rules in diagnosis and get the confidence and lift for the disease after that we integrate the results from machine learning algorithms to Dialog flow and create a chatbot for simplicity to the end-users which is different from other works that present the result in python framework.

Weighted Association Rules:

Classification and clustering have limitations, it just tells the associated disease class. In our situation, we need to dig into the details of the disease and the associated symptoms. We want to know whether the provided details are sufficient or if we need more details and what are the required details to ask back the patent.
We thought of Association rules as a way to solve the limitations of the classification and clustering algorithms.
Considering the symptoms as the antecedents and Diseases as consequents.

[Set of Symptoms] → [Disease]

Each symptom has a severity value and so we wanted to make use of that in evaluating the rules. So, we implemented weighted association rules [1].

The weighted support is evaluated as follows:

$$wsp\ (AB) = \frac{\sum_{k=1}^{|WS_T|\ \&\ (A\cup B)\subseteq t_k} weight\ (t_k)}{\sum_{k=1}^{|WS_T|} weight\ (t_k)} \qquad weight(t_k) = \frac{\sum_{i=1}^{|\bar{WS_t}(t_k)|} weight(item(i))}{|WS_t(t_k)|}$$

confidence = wsp(AB)/wsp(A)

Lift = wsp(AB)/(wsp(A)*wsp(B))

Lift [2] measures how important the rule is.
If lift = 1, it would imply that the probability of occurrence of the antecedent and that of the consequent are independent of each other.
If lift > 1, that lets us know the degree to which two occurrences are dependent on one another and makes those rules potentially useful for predicting the consequent in future data sets.

Preprocessing:

To evaluate the rules easily, we used one hot encoding mechanism to put each disease as a column beside the columns of the symptoms. The symptoms values are filled with the severity level but the severity level of the disease is not available in our data so to be unbiased we set a constant value to all the diseases and to have less effect on the rules than the symptoms we've set it to lower than the lowest symptom weight to be 0.1

Rules constructions:

Our rules are already predefined for each row we have symptoms implies disease. Then evaluating our metrics for these rules to create a data frame like follows:

Index	itemset	Disease	support	confidence	lift	ItemSetLen
0	['itching', 'skin_rash', 'nodal_skin_eruptions', 'dischromic_patches', 'Disease|Fungalinfection']	Fungalinfection	0.00257565	1	81.5849	5
1	['skin_rash', 'nodal_skin_eruptions', 'dischromic_patches', 'Disease|Fungalinfection']	Fungalinfection	0.00556688	1	81.5849	4
2	['itching', 'nodal_skin_eruptions', 'dischromic_patches', 'Disease|Fungalinfection']	Fungalinfection	0.0051102	1	81.5849	4
3	['itching', 'skin_rash', 'dischromic_patches', 'Disease|Fungalinfection']	Fungalinfection	0.00488186	1	81.5849	4
4	['itching', 'skin_rash', 'nodal_skin_eruptions', 'Disease|Fungalinfection']	Fungalinfection	0.00442519	1	81.5849	4
5	['continuous_sneezing', 'shivering', 'chills', 'watering_from_eyes', 'Disease|Allergy']	Allergy	0.00294099	1	71.4666	5
6	['shivering', 'chills', 'watering_from_eyes', 'Disease|Allergy']	Allergy	0.00570388	1	71.4666	4

Description About the rules Data:

Index	itemset	Disease	support	confidence	lift	ItemSetLen
count	304	304	304	304	304	304
unique	304	41	nan	nan	nan	nan
top	['itching', 'skin_rash', 'nodal_skin_eru…	HepatitisD	nan	nan	nan	nan
freq	1	10	nan	nan	nan	nan
mean	nan	nan	0.00619219	0.957392	41.2862	8.64803
std	nan	nan	0.00147967	0.176459	19.0399	3.60204
min	nan	nan	0.0016623	0.107346	4.66841	4
25%	nan	nan	0.00570388	1	29.9772	6
50%	nan	nan	0.00659111	1	35.6861	8
75%	nan	nan	0.00718285	1	48.9874	11
max	nan	nan	0.0082036	1	126.282	18

Now, in order to know whether the input symptoms are sufficient or not. We construct new rules containing the input symptoms as antecedents and disease from the potential diseases (a disease that contains the input symptoms) as a consequence. After that, we select the rule with the highest lift value and compare it with the lift threshold which has been set to the minimum lift value of our predefined rules approximately 4.6 (as shown in the description table above).
If the maximum lift is higher than the threshold then the rule is accepted and returns the associated disease.
If the maximum lift is lower than the threshold then, from our predefined rules we get the rule with the highest lift containing the input symptoms as part of its itemset and ask the user back if he has any of the remaining symptoms. Then we do the same processes once again but now we have two input sets one is the updated input symptoms and the other one is the symptoms that the patent doesn't have (note that at the first iteration this set was empty).

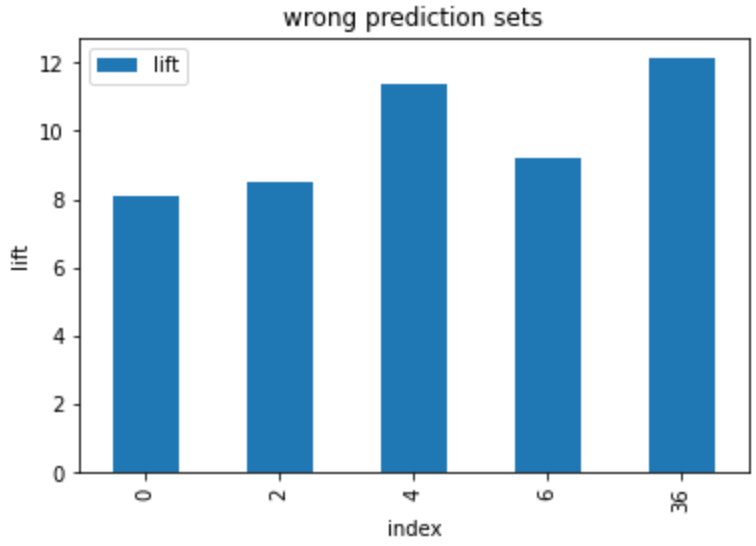
Evaluation and error analysis:

We picked one of the predefined rules.

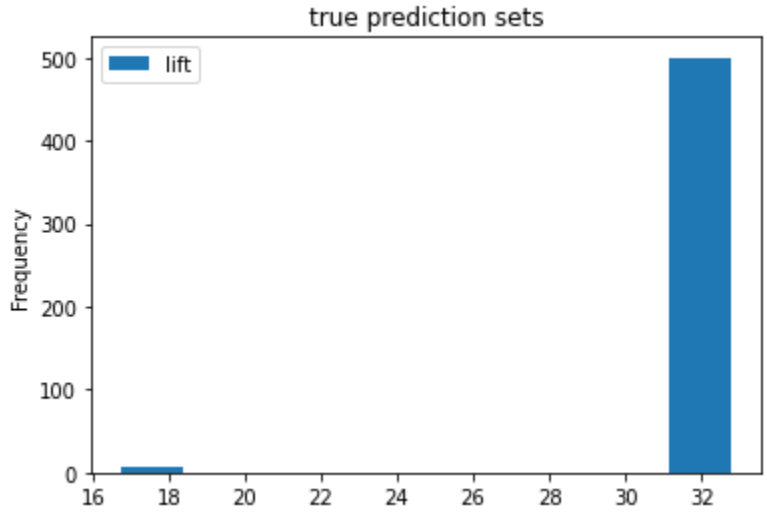
Index	itemset	Disease	support	confidence	lift	itemSetLen
0	['weight_loss', 'restlessness', 'lethargy', 'irregular_sugar_level', 'blurred_and_distorted_vision', 'obesity', 'excessive_hunger', 'increased_appetite', 'polyuria', 'Disease|Diabetes']	Diabetes	0.00680115	1	32.7645	10

The item set contains 9 symptoms and 1 disease. From those 9 symptoms, we generated all possible combinations from 1 item to 9 and so we generated 511 input symptoms sets. Then we tried to predict the disease.
Column names are:
["inputset","prediction","lift","confidence","predicted_diease","actual_disease","num_remaining_symptoms","num_potential_disease"]

Index	inputset	prediction	lift	confidence	predicted_diease	actual_disease	num_remaining_symptoms	num_potential_disease
0	['weight_loss']	False	8.06236	0.293852	Tuberculosis	Diabetes	7	4
1	['restlessness']	True	16.7569	0.511435	Diabetes	Diabetes	10	2
2	['lethargy']	False	8.5121	0.284354	Chickenpox	Diabetes	10	4
3	['irregular_sugar_level']	True	32.7645	1	Diabetes	Diabetes	9	1
4	['blurred_and_distorted_vision']	False	11.3503	0.329193	Migraine	Diabetes	7	3
5	['obesity']	True	17.0061	0.519041	Diabetes	Diabetes	6	2
6	['excessive_hunger']	False	9.20103	0.271215	Hypoglycemia	Diabetes	7	4
7	['increased_appetite']	True	32.7645	1	Diabetes	Diabetes	9	1
8	['polyuria']	True	32.7645	1	Diabetes	Diabetes	9	1



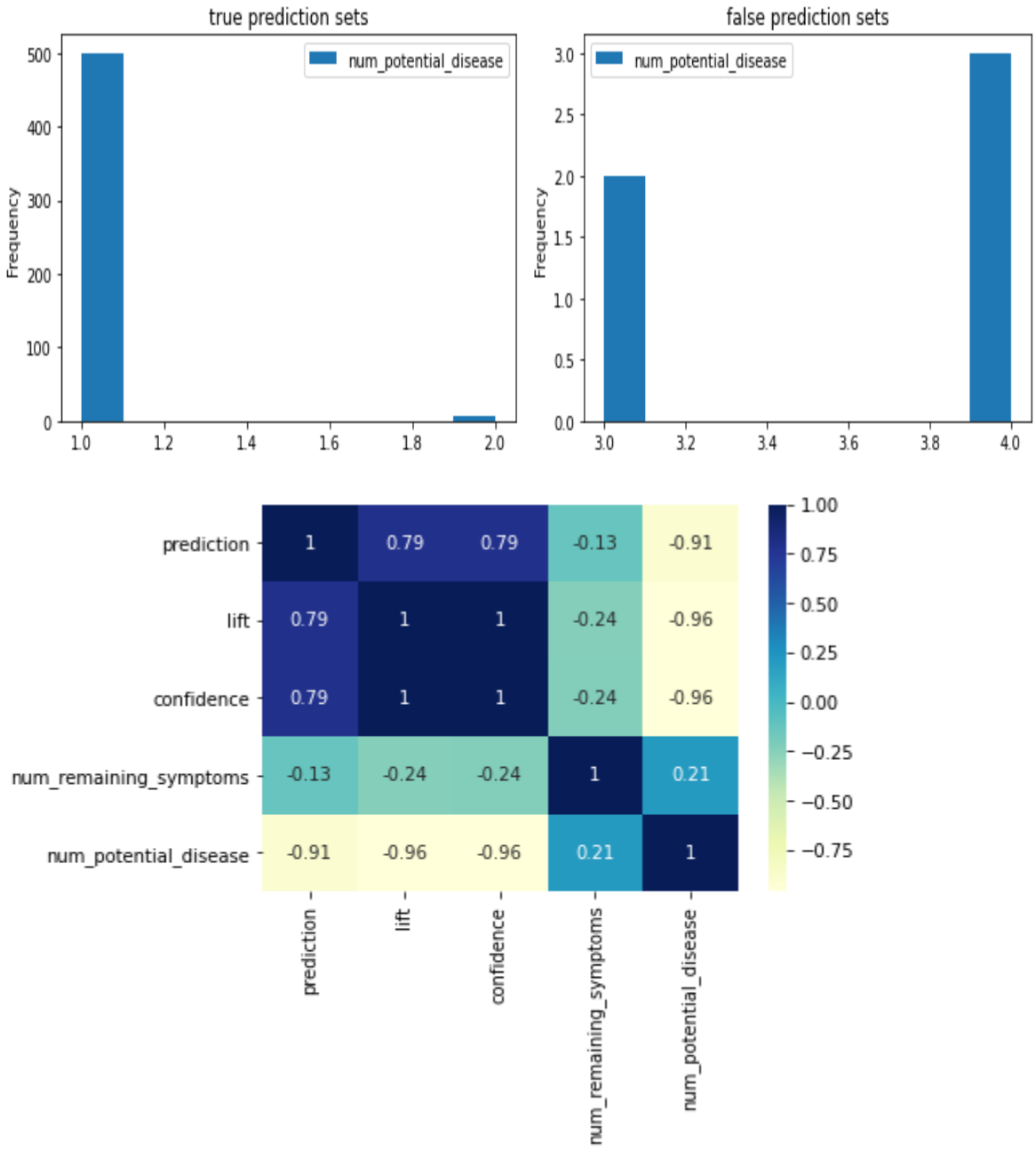
Can you see that all the input sets passed the minimum lift threshold (~4) and yet the prediction was wrong. Maybe we need to reevaluate our threshold again.



From the truly predicted sets' frequency plot we see that most of them are concentrated at lift = 32 which is the same value as the original rule of our disease.

Index	itemset	Disease	support	confidence	lift	ItemSetLen
0	['weight_loss', 'restlessness', 'lethargy', 'irregular_sugar_level', 'blurred_and_distorted_vision', 'obesity', 'excessive_hunger', 'increased_appetite', 'polyuria', 'Disease|Diabetes']	Diabetes	0.00680115	1	32.7645	10

Few of the symptoms sets have a lift value of around 16.7 which equals the value of the original lift value divided by the number of the potential diseases which is 2 (32.7645/2 = 16.38225).



Can you feel it now?

We conclude that the lift and the number of potential diseases are highly correlated. In fact, the number of potential diseases is more correlated with the prediction than the lift value, we can replace our lift threshold with the number of potential diseases threshold. We couldn't benefit from the Weighted Association Rules in our case, but imagine that we have more valuable information in the data like the symptoms' priority and importance concerning each disease, also the disease severity and more information could make a huge difference and at that time we can see the high impact of the Weighted Association Rules.

We are looking forward to collecting and extracting more valuable information and gathering a higher-quality dataset. With the help of doctors and other sources of information like books.

References:

[1] Tao, F., Murtagh, F., & Farid, M. (n.d.). *Weighted Association rule mining using weighted support and Significance Framework*. Retrieved August 2, 2022, from https://eprints.soton.ac.uk/257986/1/331.tao.pdf

[2] *Association rules*. NoSimpler. (n.d.). Retrieved August 3, 2022, from https://www.nosimpler.me/association-rules/

Dataset resources

https://www.cancer.gov/publications/dictionaries/cancer-terms/def/diagnosis

https://en.wikipedia.org/wiki/Prognosis

https://www.cancer.gov/about-cancer/diagnosis-staging/diagnosis