Early Diagnosis Digital Medical Assistant

Andrea Capella-Castro, Angie Menjivar, Francesco Coccaro, Nicole Gutierrez and Patrick Kelly

Agenda

- Meet the Team
- The Problem + Our Ideas
- Objective & Hypothesis
- Data Approach
- Dataset Overview + Process
- Variables
- Demo/Code Walkthrough
- Symptom Frequency + Importance
- Correlation Between Importance
- Log Loss
- Conclusions
- Recommendations
- Solutions
- App Demo

Meet the Team

1. Andrea Capella-Castro - Presentation Leader
2. Angie Menjivar - Project Leader
3. Francesco Coccaro - Project Leader
4. Patrick Kelly - Technical Leader
5. Nicole Gutierrez - Technical Leader

The Problem

According to a study by PinnacleCare, diagnostic errors and other inefficiencies cost the U.S. economy $750 billion each year. This includes:

$200 billion in unnecessary healthcare costs
$250 billion in lost productivity
$300 billion in pain and suffering

The study found that diagnostic errors are most common in the following areas:

Cancer
Heart disease
Stroke
Mental illness
Chronic pain

How this idea came to be...

The development of LLMs has allowed super-symbolic representation of information into symbolic and subsymbolic meaning representing knowledge domains

This empowers patients to become more aware of Healthcare discrepancies/inequalities through the early diagnosis process

Make something accessible / affordable for people

Closing the knowledge gap between patients and healthcare providers

Objective

The objective is to build a model that could predict a disease from symptoms provided by patients.

Hypothesis

H0: Diseases cannot be predicted based on symptom input.

HA: Diseases can be predicted based on symptom input.

Data approach

- Data Source: **Kaggle**
- Data Cleaning: **Excel**
- Analytics + Insights: **H2O Flow Platform, Excel**
- Machine Learning and Predictive Analytics: **Python**

Dataset Overview

- Our **Dataset** consists **diseases** and **symptoms**

- 131 symptoms
- 41 diseases
- 120 cases per disease
  - original: 18 columns x 4921 rows
  - final: 133 columns x 4921 rows

Dataset Process

Data processing problems-

1. <u>Reformatting matrix</u>
2. <u>Data size/Finding Data</u>
3. <u>Narrow variable set</u>

Original CSV

18 columns x 4921 rows

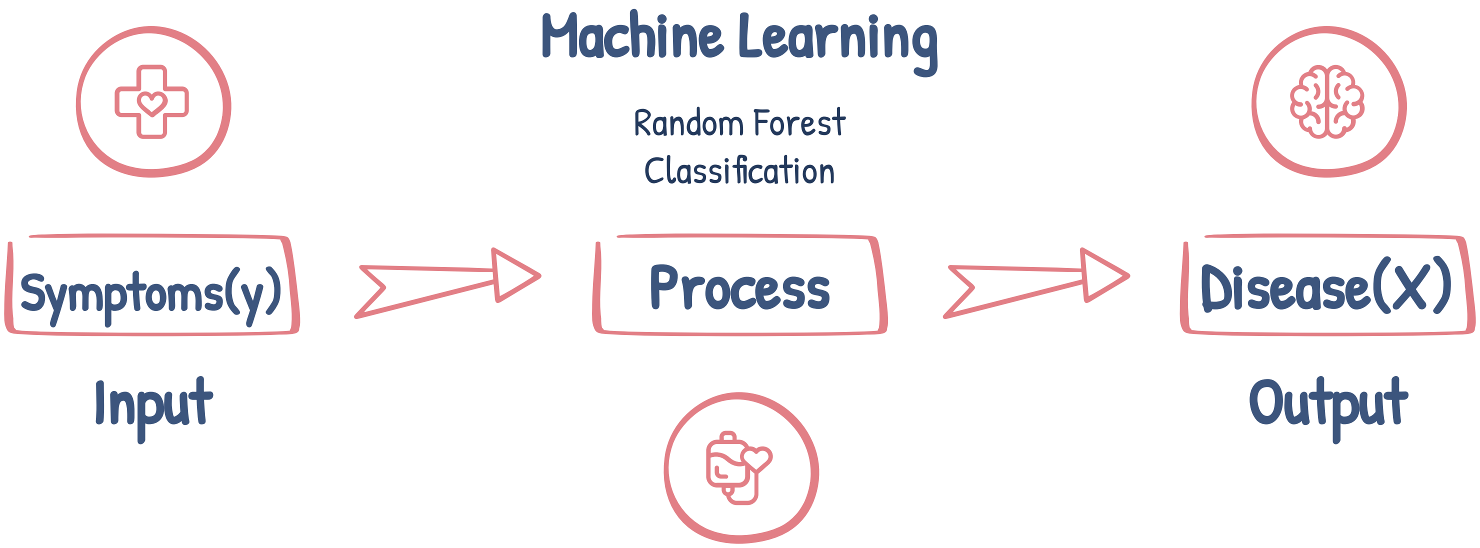
Disease	Symptom_1	Symptom_2	Symptom_3
Fungal infection	itching	skin rash	nodal skin eruptions
Fungal infection	skin rash	nodal skin eruptions	dischromic patches
Fungal infection	itching	nodal skin eruptions	dischromic patches
Fungal infection	itching	skin rash	dischromic patches

Confusion Matrix

133 columns x 4921 rows

Patient	abdominal pain	abnormal menstruation	acidity	Disease
1	0	0	0	Fungal infection
2	0	0	0	Fungal infection
3	0	0	0	Fungal infection
4	0	0	0	Fungal Infection

Variables



Machine Learning
Random Forest
Classification
Symptoms(y)
Process
Disease(X)
Input
Output

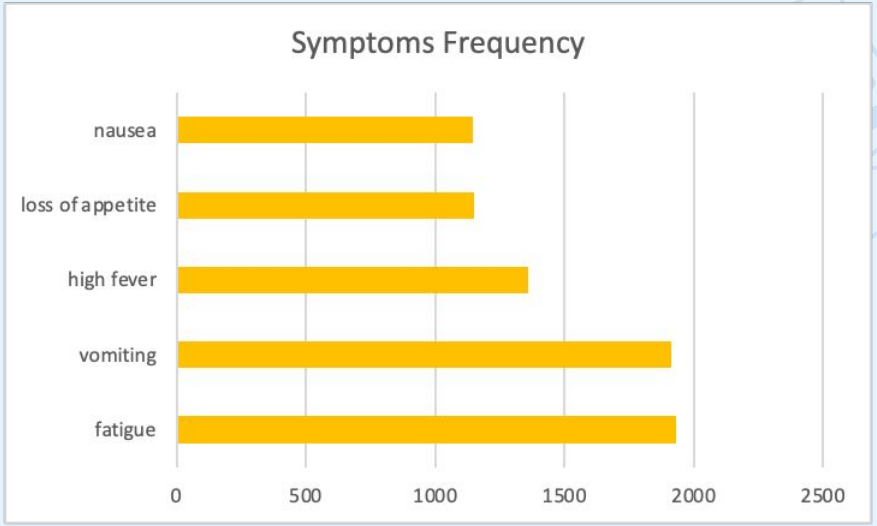
Demo/Code walkthrough

```
# Step 1: Data Preparation
X = df.drop(columns=['Disease'])  # Features (symptoms)
y = df['Disease']  # Target variable (disease)

# Split the data into training and testing sets (80% training, 20% testing)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 2: Model Selection and Training
model = RandomForestClassifier(random_state=42)
```

```
Enter a comma-separated list of symptoms (e.g., itching,sweating,vomiting): itching,sweating,vomiting
Predicted Disease: Heart attack
P-values and Significant Diseases:
Significant Disease: Heart attack P-value: 0.041833091688969676
Correlating Symptoms to Predicted Disease:
['breathlessness', 'chest pain', 'sweating', 'vomiting']
```

Frequency of symptoms

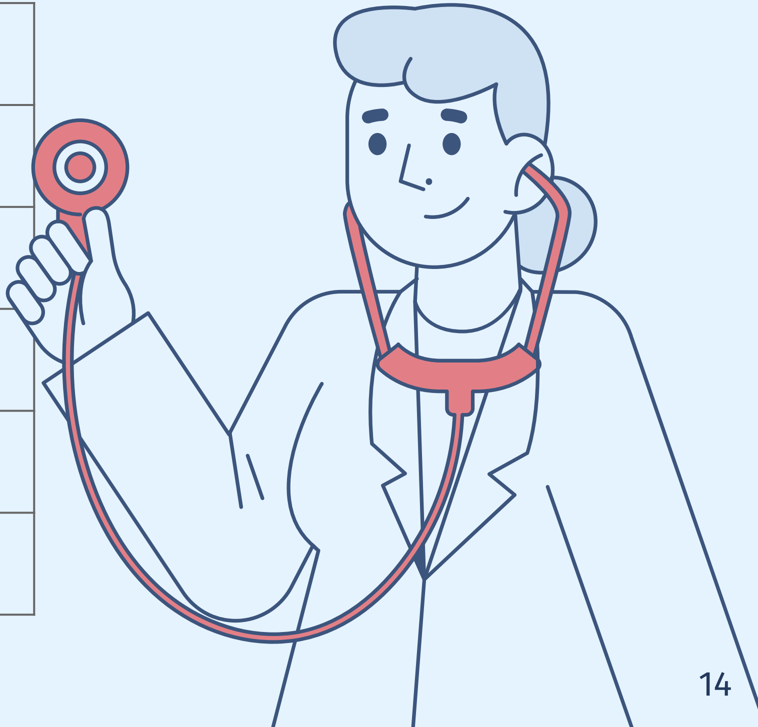
Symptom	Frequency
fatigue	1932
vomiting	1914
high fever	1362
loss of appetite	1152
nausea	1146



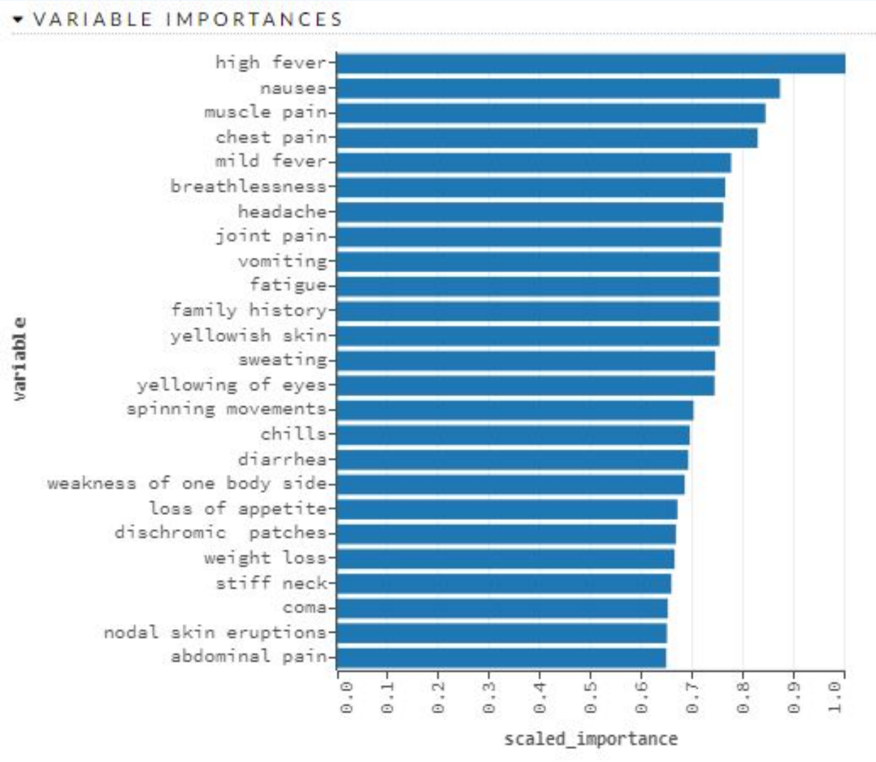
Symptom Importance

Top 5 Highest Symptom Means Relating to Diseases

Symptom	Mean
High fever	0.2768
Loss of appetite	0.2341
Nausea	0.2329
Abdominal pain	0.2098
Yellowish skin	0.1854



Variable Importance



Greatest Importance

variable	relative_importance	scaled_importance	percentage
high fever	2566.7666	1.0	0.0166
nausea	2236.1882	0.8712	0.0145
muscle pain	2162.7761	0.8426	0.0140
chest pain	2123.5366	0.8273	0.0137
mild fever	1989.4247	0.7751	0.0129

Least Importance

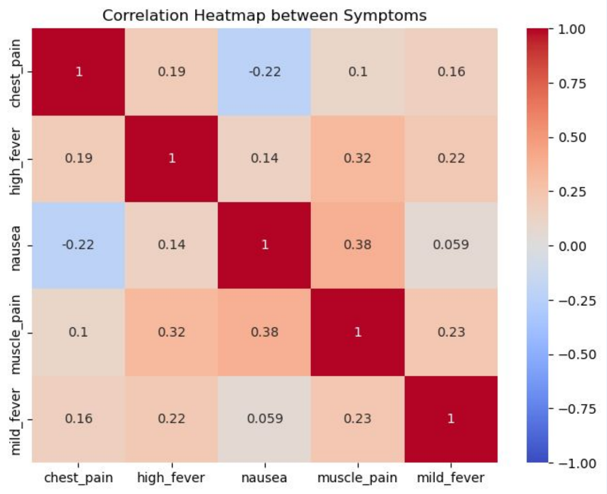
pain during bowel movements	457.7419	0.1783	0.0030
painful walking	437.2794	0.1704	0.0028
puffy face and eyes	423.2812	0.1649	0.0027
drying and tingling lips	411.5307	0.1603	0.0027
cramps	368.7041	0.1436	0.0024

The higher the importance, the stronger influence on model predictions

Correlation between most important variables

variable	relative_importance	scaled_importance	percentage
high fever	2566.7666	1.0	0.0166
nausea	2236.1882	0.8712	0.0145
muscle pain	2162.7761	0.8426	0.0140
chest pain	2123.5366	0.8273	0.0137
mild fever	1989.4247	0.7751	0.0129

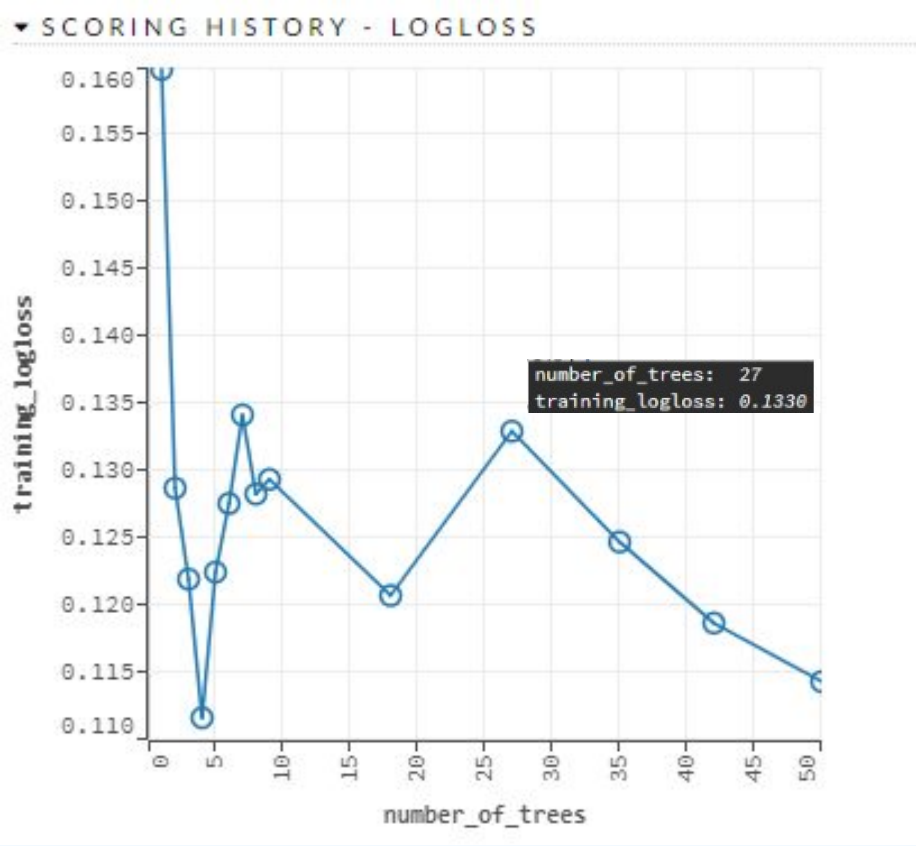
We chose to analyze the most important variables in the dataset



Top 3 Correlations

nausea	muscle_pain	0.377418
muscle_pain	high_fever	0.315278
mild_fever	muscle_pain	0.228976

Log Loss



Scoring history typically shows how the performance metrics (such as log loss) of your model change over iterations or epochs during the training process. Log loss is a common evaluation metric for classification problems, especially when dealing with probability predictions. **Lower log loss values indicate better model performance.**

Conclusions

- Insights, Variable importance and limitations
- We are able to make a disease predictive model based on symptoms, but not with this dataset
  Reject our null hypothesis (HA: Diseases can be predicted based on symptom input)
- How can this dataset helped for future recommendations, connect with early diagnostic platform

Recommendations

Prescriptions for a similar projects

- We would recommend this process to a hospital to gain better insights on early diagnosis
- Bottleneck of our process was collecting and cleaning of data...
- More data would increase accuracy
- Diverse data will improve use case and breadth of prediction (Do more with more)

Solutions

How to find the disease?

1st — **Complexity**

Creating a more complex dataset (adding more attributes to gain better insights)

2nd — **Modeling**

Implement more modeling from updated dataset

3rd — **Collection**

Creating a data collection process

App demo

Thank you!