Machine Learning: Disease Prediction

- Athen Osterberg
- Heather Shoberg
- Mi Thao
- Lori Vitaioli

Objective

Train a machine learning model to predict what **disease** a patient might have based on their **symptoms**

Steps

1. Determine which data set to use and pre-process the data

2. Develop four machine learning models to make predictions of diseases based on symptoms

3. Optimize models to reach 75%+ accuracy

4. Create a “Disease Predictor” application using HTML, flask, and a machine learning model that can predict a user’s potential disease based on the symptoms they input

Data Set

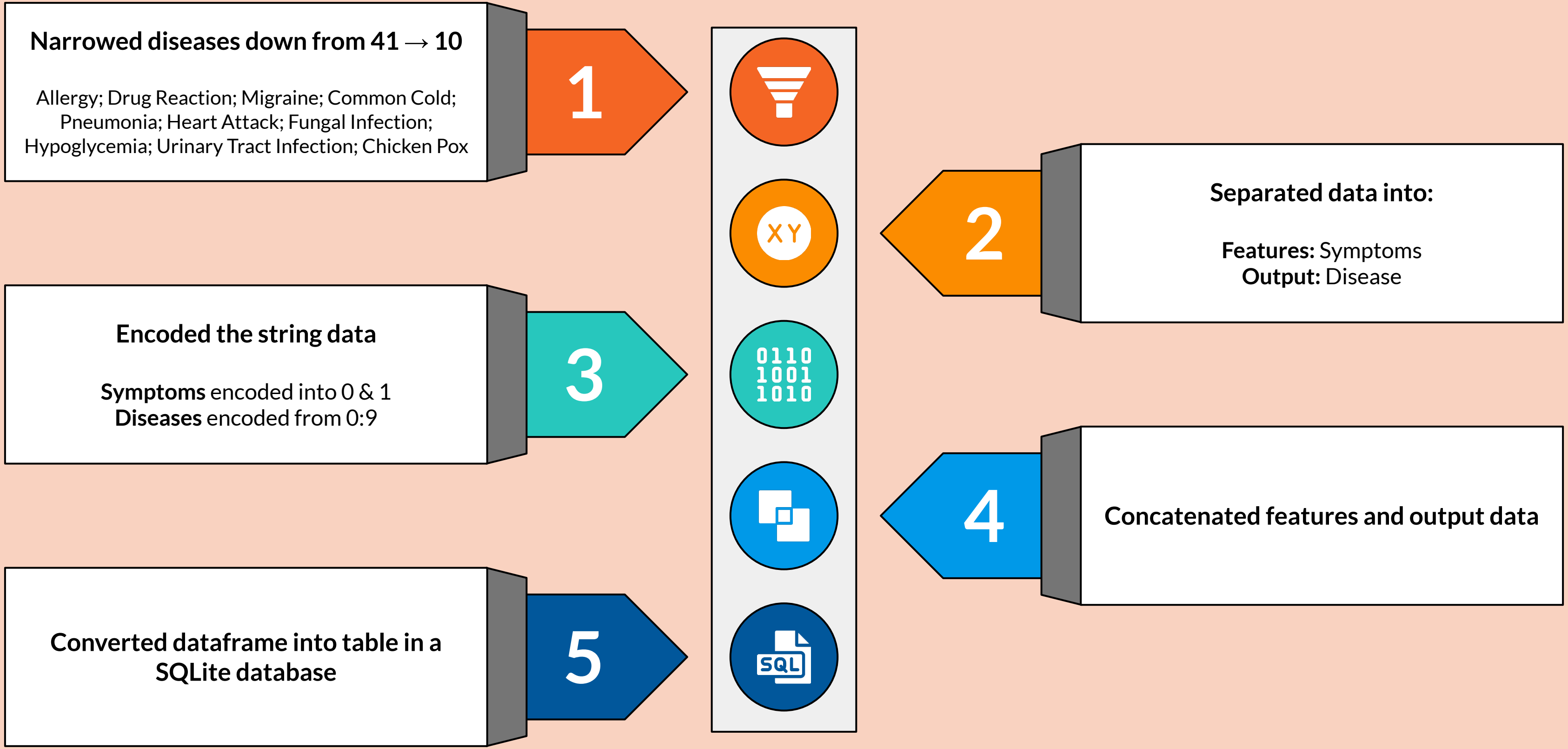
Kaggle:
Disease Symptom Prediction

41 Diseases

120 Rows of Symptom Data per Disease

Up to 17 Symptoms per Disease

Data Pre-Processing



1

Narrowed diseases down from 41 → 10

Allergy; Drug Reaction; Migraine; Common Cold; Pneumonia; Heart Attack; Fungal Infection; Hypoglycemia; Urinary Tract Infection; Chicken Pox

2

Separated data into:

Features: Symptoms
Output: Disease

3

Encoded the string data

Symptoms encoded into 0 & 1
Diseases encoded from 0:9

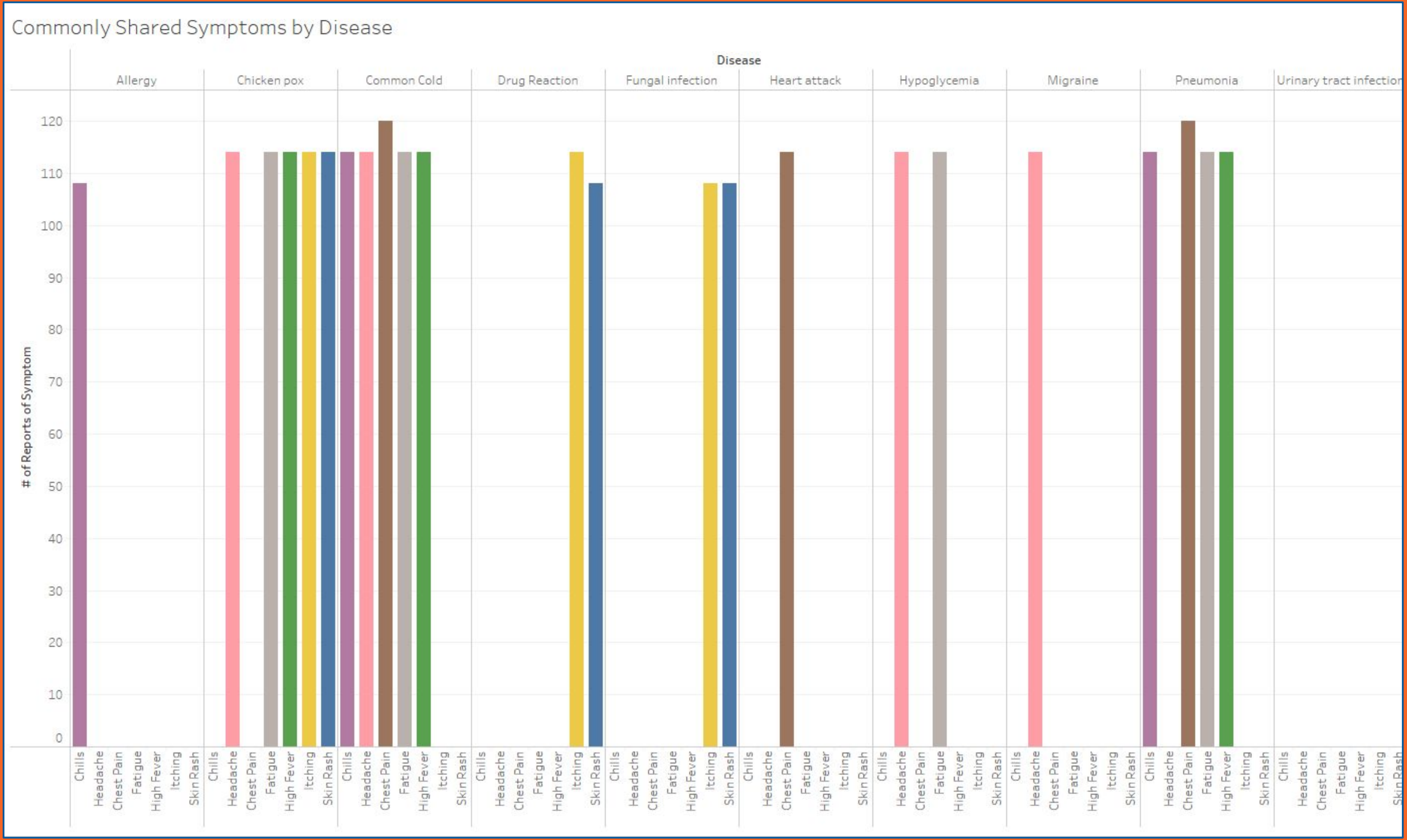
4

Concatenated features and output data

5

Converted dataframe into table in a SQLite database

Initial Data Analysis

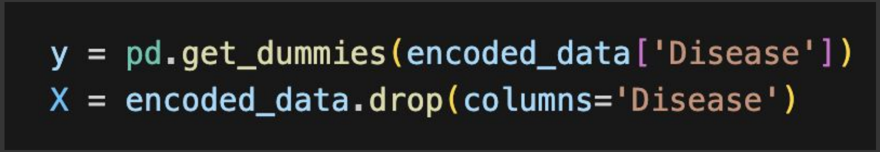


51
Total Symptoms

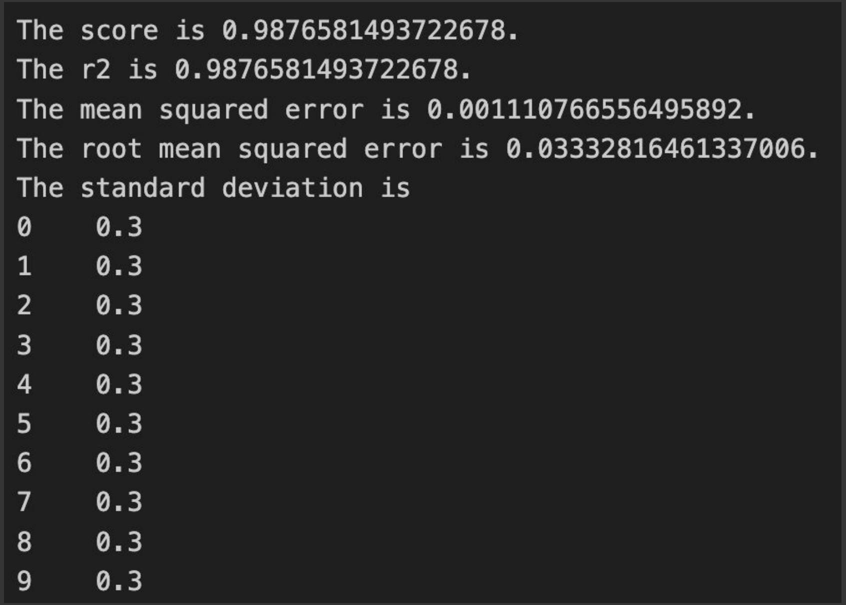
7
Symptoms Compared

ML #1 - Linear Regression

- X values are symptoms
- y value is the diseases column
- Y needs to be split into dummies since it is categorical
- Fit a model using sklearn linear regression



```
y = pd.get_dummies(encoded_data['Disease'])
X = encoded_data.drop(columns='Disease')
```



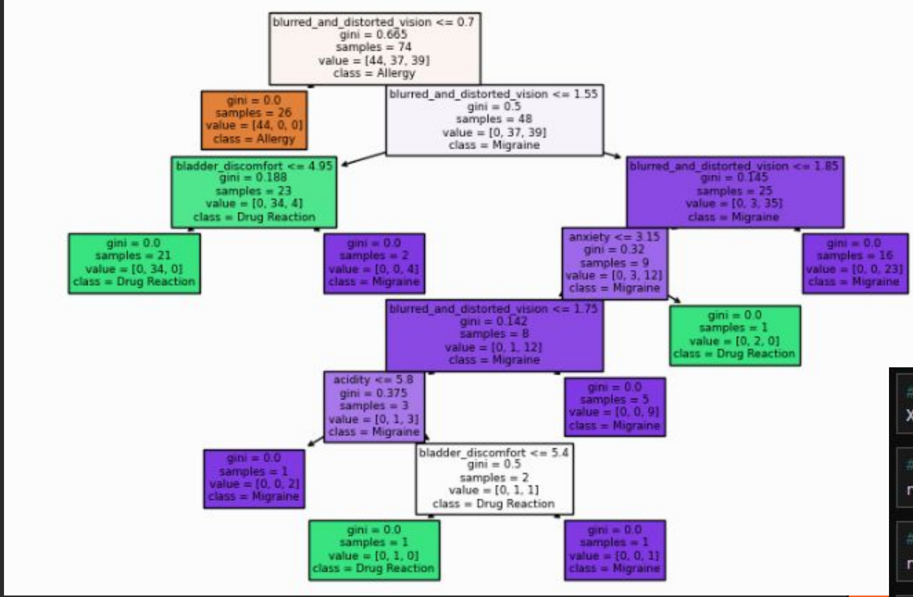
```
The score is 0.9876581493722678.
The r2 is 0.9876581493722678.
The mean squared error is 0.001110766556495892.
The root mean squared error is 0.03332816461337006.
The standard deviation is
0    0.3
1    0.3
2    0.3
3    0.3
4    0.3
5    0.3
6    0.3
7    0.3
8    0.3
9    0.3
```

ML #2 - Logistic Regression

Disease Value	Disease	Precision	Recall	F1-Score	Support
Disease 0	Allergy	1.00	1.00	1.00	21
Disease 1	Chicken Pox	1.00	1.00	1.00	23
Disease 2	Common Cold	1.00	1.00	1.00	24
Disease 3	Drug Reaction	1.00	1.00	1.00	23
Disease 4	Fungal Infection	1.00	1.00	1.00	23
Disease 5	Heart Attack	1.00	1.00	1.00	33
Disease 6	Hypoglycemia	1.00	1.00	1.00	19
Disease 7	Migraine	1.00	1.00	1.00	22
Disease 8	Pneumonia	1.00	1.00	1.00	24
Disease 9	Urinary tract infection	1.00	1.00	1.00	25
--					
	Accuracy	-	-	1.00	240
	Macro Avg	1.00	1.00	1.00	240
	Weighted Avg	1.00	1.00	1.00	240

The classification report shows strong performance, with precision, recall, and F1-scores of 1.00 across all diseases. This combined with the overall accuracy of 1.00 indicates high accuracy in identifying diseases based on the dataset of symptoms used.

— ML #3 - Random Forest

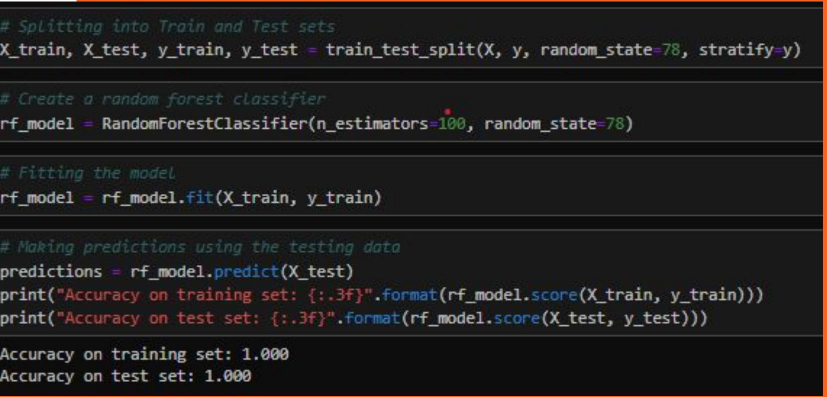


*Example decision tree from random forest model featuring blurred and distorted vision symptom

Training/Testing sets based on 10 random diseases and their symptoms.

RF Classifier n_estimators 10 gave a 99% level of accuracy

RF Classifier n_estimators 100 gave a 100% level of accuracy



```
# Splitting into Train and Test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=78, stratify=y)

# Create a random forest classifier
rf_model = RandomForestClassifier(n_estimators=100, random_state=78)

# Fitting the model
rf_model = rf_model.fit(X_train, y_train)

# Making predictions using the testing data
predictions = rf_model.predict(X_test)
print("Accuracy on training set: {:.3f}".format(rf_model.score(X_train, y_train)))
print("Accuracy on test set: {:.3f}".format(rf_model.score(X_test, y_test)))
```

```
Accuracy on training set: 1.000
Accuracy on test set: 1.000
```

ML #4 - Neural Network

Version 1

Layers Breakdown:

- **1st Layer:**
  - 10 Input Dimensions
  - 10 Nodes
  - Relu Activation Function
- **2nd Layer:**
  - 10 Nodes
  - Relu Activation Function
- **Output Layer:**
  - 10 Nodes
  - Softmax Activation Function

Accuracy: 98.3%

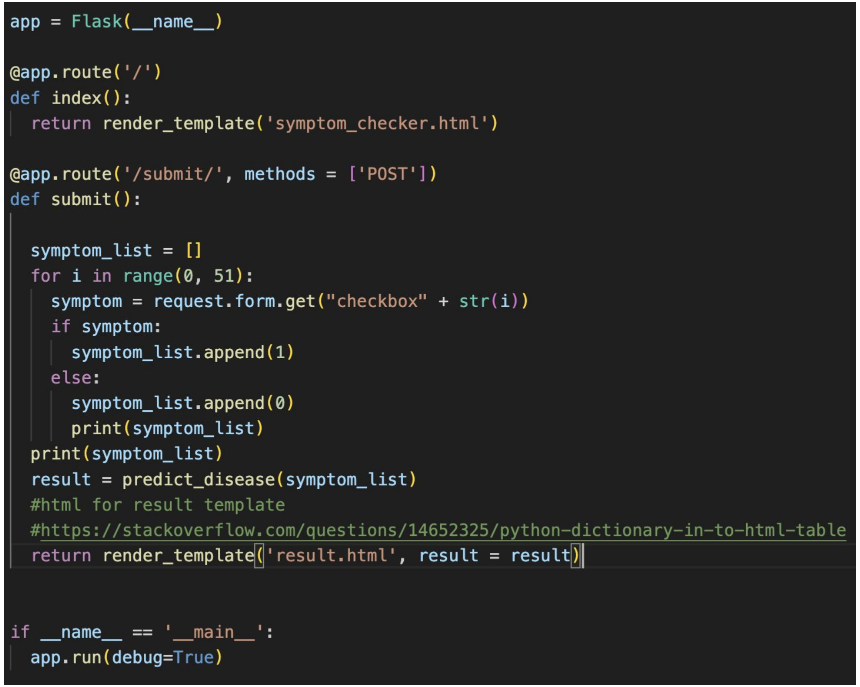
Version 2

Layers Breakdown:

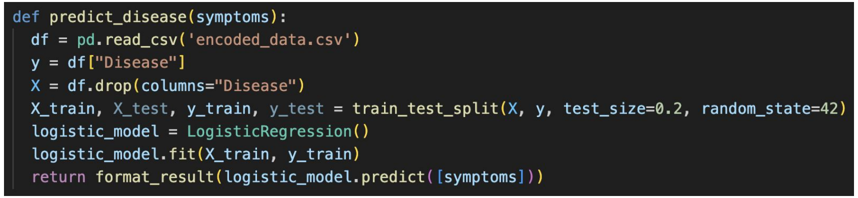
- **1st Layer:**
  - 10 Input Dimensions
  - 20 Nodes
  - Relu Activation Function
- **2nd Layer:**
  - 10 Nodes
  - Relu Activation Function
- **Output Layer:**
  - 10 Nodes
  - Softmax Activation Function

Accuracy: 100%

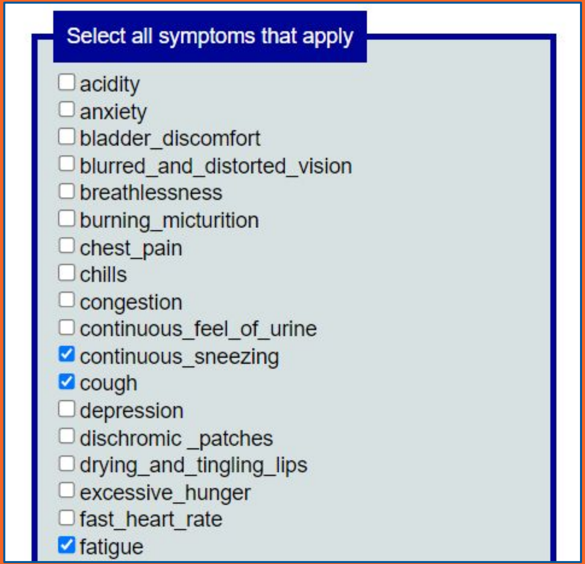
Disease Prediction Flask API



```
app = Flask(__name__)

@app.route('/')
def index():
  return render_template('symptom_checker.html')

@app.route('/submit/', methods = ['POST'])
def submit():

  symptom_list = []
  for i in range(0, 51):
    symptom = request.form.get("checkbox" + str(i))
    if symptom:
      symptom_list.append(1)
    else:
      symptom_list.append(0)
      print(symptom_list)
  print(symptom_list)
  result = predict_disease(symptom_list)
  #html for result template
  #https://stackoverflow.com/questions/14652325/python-dictionary-in-to-html-table
  return render_template('result.html', result = result)


if __name__ == '__main__':
  app.run(debug=True)
```

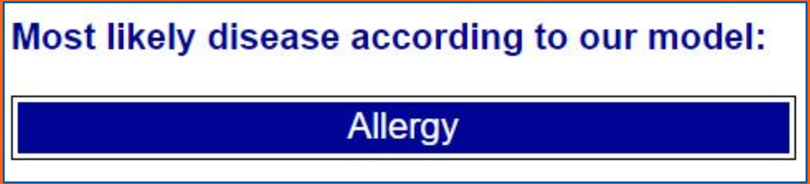


```
def predict_disease(symptoms):
  df = pd.read_csv('encoded_data.csv')
  y = df["Disease"]
  X = df.drop(columns="Disease")
  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
  logistic_model = LogisticRegression()
  logistic_model.fit(X_train, y_train)
  return format_result(logistic_model.predict([symptoms]))
```



Select all symptoms that apply

☐ acidity
☐ anxiety
☐ bladder_discomfort
☐ blurred_and_distorted_vision
☐ breathlessness
☐ burning_micturition
☐ chest_pain
☐ chills
☐ congestion
☐ continuous_feel_of_urine
☑ continuous_sneezing
☑ cough
☐ depression
☐ dischromic _patches
☐ drying_and_tingling_lips
☐ excessive_hunger
☐ fast_heart_rate
☑ fatigue



Most likely disease according to our model:

Allergy

Conclusion & Next Steps

- ➔ **¾ of the machine learning models achieved 100% accuracy**

  The models were able to easily interpret the dataset due to the consistency of symptoms for each disease

- ➔ **Limitations**

  Only selecting 10/41 diseases in the dataset means the model is limited in what it can predict

- ➔ **What's next?**

  Add more data to the model: the other 31 diseases not initially included and more beyond those

  Machine learning models like these can assist medical professionals when diagnosing patients, which can lead to more efficient treatment. However, a model alone should not be used as a diagnosis tool

Questions?