AI in Healthcare

Samaresh Maiti

Heritage Institute of Technology, Kolkata

Date: 12th January 2023

Author Note

An ideation report for Feynn Lab

Abstract

Artificial Intelligence is gradually changing medical practices and their approach to various problems. With recent progress in digitized data acquisition, machine learning and computing infrastructure, AI applications are expanding into areas that were previously thought to be only the province of human experts. According to the CDC, heart disease is one of the leading causes of death for people and requires early attention. To avoid mishaps is better to be safe than sorry. The main objective of this report is to apply Machine Learning algorithms for the prediction and provide faster and highly accurate results.

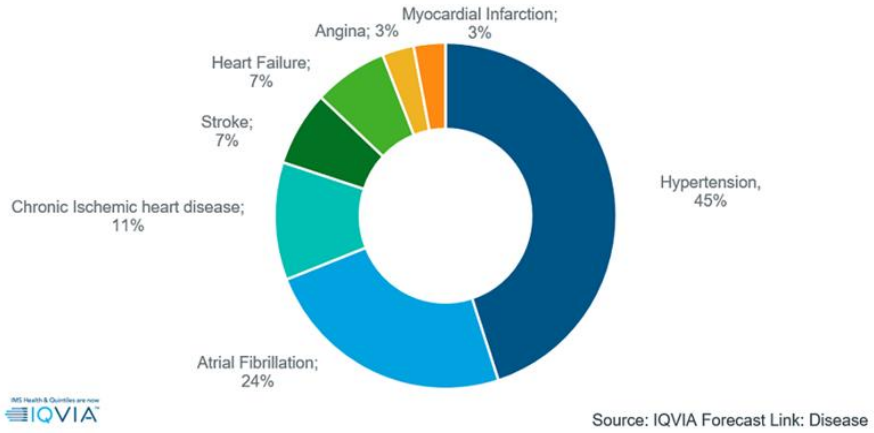
Keywords: Artificial Intelligence, Medical, Heart Disease, Machine Learning

1. Problem Statement

Cardiovascular diseases (CVDs) are the leading cause of death globally, taking an estimated 17.9 million lives each year. CVDs are a group of disorders of the heart and blood vessels and include coronary heart disease, cerebrovascular disease, rheumatic heart disease and other conditions. Identifying those at the highest risk of CVDs and ensuring they receive appropriate treatment can prevent premature deaths. To identify such conditions early machine-learning algorithms can be used which not only provide quick results but also are highly accurate. Hence incorporating machine learning in health care can greatly benefit the field as well as people.

2. Market / Customer/ Business Need Assessment:

Cardiovascular Diseases (CVD) are a group of disorders affecting the heart and blood vessels and are the most common cause of death globally. According to WHO1, CVD contributed to approximately 31% or 17.9 million death globally in the year 2016. According to Forecast Link, the market for major heart disease prescription drug sales contributed $67 billion in 2018 and has only grown at a CAGR of 0.6% since 2014, mainly due to the significant generic erosion of a blockbuster hypertension brand from late 2013. However, going forward, this market is expected to grow at a CAGR of 5.5% (2018-24).



It is known that customers like fast, accurate and hassle-free services. It has been observed that early identification of heart disease can prevent complications such as heart failure, stroke, kidney disease and artery disease. The tests for heart disease take a lot of time and many require at least 12 hours of fasting to get the final result. With the rapid growth of Artificial Intelligence, it is now possible to do tests more efficiently and effortlessly. With the boon of a huge amount of data, it is now possible for machines to work hand in hand with medical personnel to provide better services.

In this report, I am going to emphasize about Machine Learning approach, which is a branch of artificial intelligence, in the early identification of heart disease and preventing of serious complications. This system takes in data, processes, trains itself to identify patterns, and provides outcomes. This report explores the business idea that can be derived from the application at the backend.

3. Target specifications and Characterization

- Provide users with hassle-free, quick and accurate results of heart tests.
- Develop a business model by connecting with doctors and hospitals.

The above-mentioned targets can be achieved by analyzing:

1. The input dataset.
2. The patients' condition.
3. Problems faced by patients suffering from heart disease.
4. Refer potential patients to tie up hospitals and doctors.
5. Reminding users of healthy habits and ways of taking good care of their hearts.

4. External Searches (Information Searches).

For this report, the 'heart_2020_cleaned.csv' dataset has been used from Kaggle on which the working prototype is to be developed. The data set consists of 319795 rows x 18 columns. The 18 columns contain several attributes related to CVD such as BMI, Physical Health, Mental Health, Diabetic and many more. The overview of the dataset is as follows:

```python
import pandas as pd

heart_data=pd.read_csv('heart_2020_cleaned.csv')
heart_data.head()
```

✓ 0.8s Python

	HeartDisease	BMI	Smoking	AlcoholDrinking	Stroke	PhysicalHealth	MentalHealth	DiffWalking	Sex	AgeCategory	Race	Diabetic	PhysicalActivity	G
0	No	16.60	Yes	No	No	3.0	30.0	No	Female	55-59	White	Yes	Yes	
1	No	20.34	No	No	Yes	0.0	0.0	No	Female	80 or older	White	No	Yes	
2	No	26.58	Yes	No	No	20.0	30.0	No	Male	65-69	White	Yes	Yes	
3	No	24.21	No	No	No	0.0	0.0	No	Female	75-79	White	No	No	
4	No	23.71	No	No	No	28.0	0.0	Yes	Female	40-44	White	No	Yes	

```python
heart_data.info()
```

✓ 0.5s

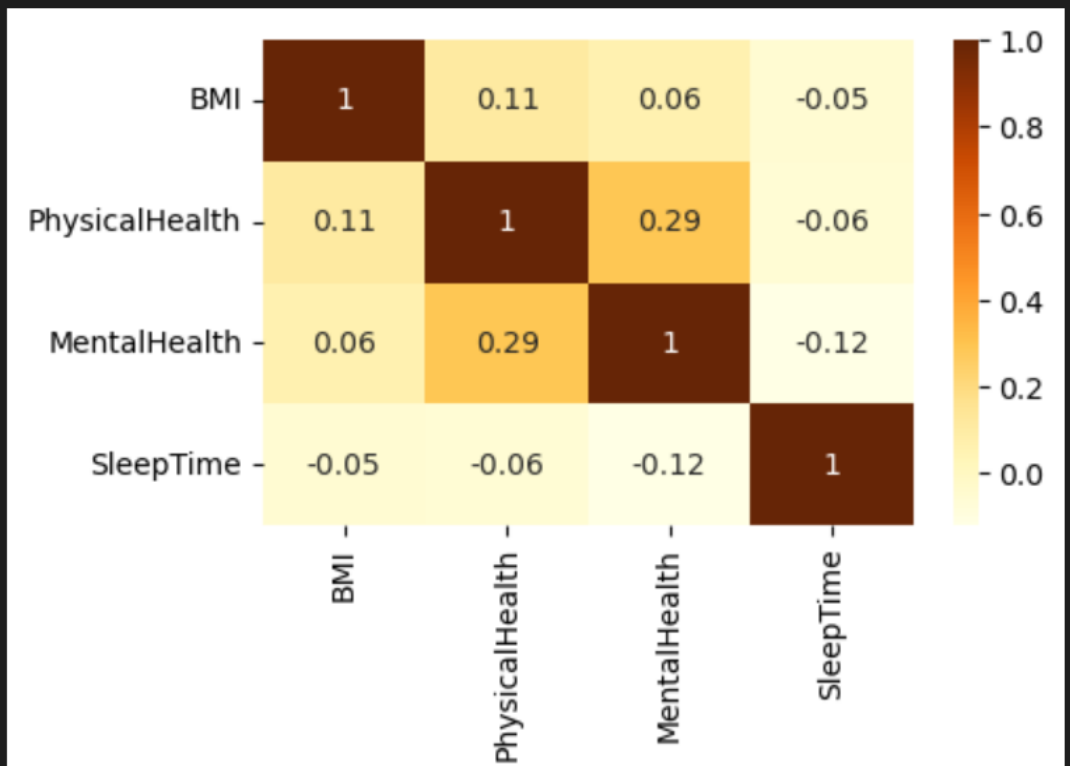
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 319795 entries, 0 to 319794
Data columns (total 18 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   HeartDisease      319795 non-null  object
 1   BMI               319795 non-null  float64
 2   Smoking           319795 non-null  object
 3   AlcoholDrinking   319795 non-null  object
 4   Stroke            319795 non-null  object
 5   PhysicalHealth    319795 non-null  float64
 6   MentalHealth      319795 non-null  float64
 7   DiffWalking       319795 non-null  object
 8   Sex               319795 non-null  object
 9   AgeCategory       319795 non-null  object
 10  Race              319795 non-null  object
 11  Diabetic          319795 non-null  object
 12  PhysicalActivity  319795 non-null  object
 13  GenHealth         319795 non-null  object
```

5. Benchmarking

```
1 import seaborn as sns
2 import matplotlib.pyplot as plt
3 correlation=heart_data.corr().round(2)
4 plt.figure(figsize=(5,3))
5 sns.heatmap(correlation, annot = True, cmap = 'YlOrBr')
6
```

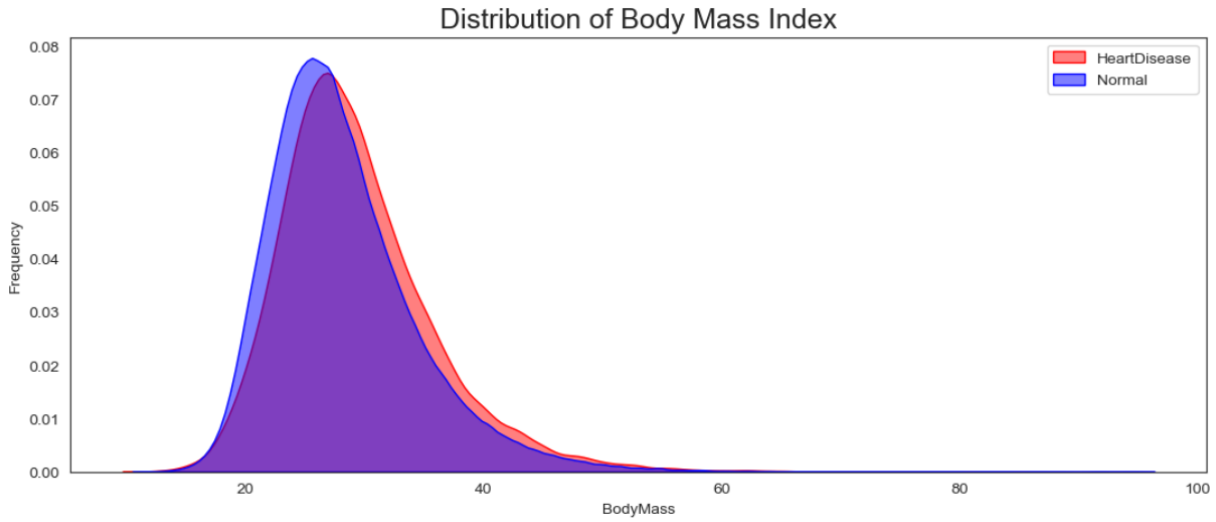
✓ 0.3s

<AxesSubplot:>

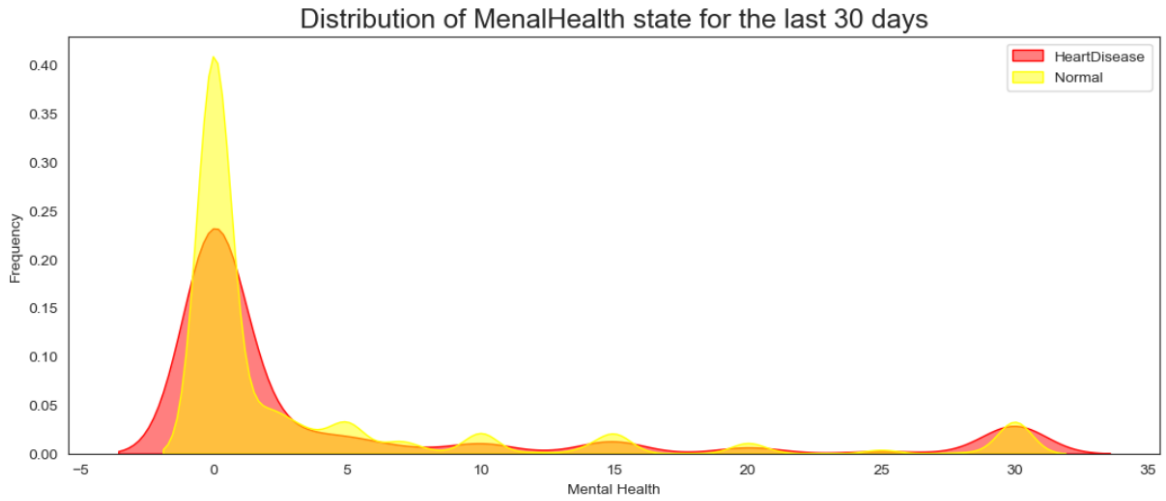



```
1 fig, ax = plt.subplots(figsize = (13,5))
2 sns.kdeplot(heart_data[heart_data["HeartDisease"]=="Yes"]["BMI"], alpha=0.5,shade = True, color="red", label="HeartDisease", ax = ax)
3 sns.kdeplot(heart_data[heart_data["HeartDisease"]=="No"]["BMI"], alpha=0.5,shade = True, color="blue", label="Normal", ax = ax)
4 plt.title('Distribution of Body Mass Index', fontsize = 18)
5 ax.set_xlabel("BodyMass")
6 ax.set_ylabel("Frequency")
7 ax.legend()
8 plt.show()
```

✓ 2.5s




```
fig, ax = plt.subplots(figsize = (13,5))
sns.kdeplot(heart_data[heart_data["HeartDisease"]=="Yes"]["MentalHealth"], alpha=0.5,shade = True, color="red", label="HeartDisease", ax = ax)
sns.kdeplot(heart_data[heart_data["HeartDisease"]=="No"]["MentalHealth"], alpha=0.5,shade = True, color="yellow", label="Normal", ax = ax)
plt.title('Distribution of MenalHealth state for the last 30 days', fontsize = 18)
ax.set_xlabel("Mental Health")
ax.set_ylabel("Frequency")
ax.legend()
plt.show()
```



The above data gives us a glimpse of the data distribution and the correlations of different attributes which contribute to the final outcome.

6. Applicable Patents.

Keerthi Kodithuwakku is a biomedical engineer, and the co-founder and CEO of Jendo Innovations, a bio-medical startup delivering patented healthcare solutions detecting abnormalities in the cardiovascular system and seeking to prevent the risk of cardiovascular diseases.

7. Applicable Regulations.

The patents mentioned above might claim the technology used if the algorithms are not developed and optimised individually and for our requirements. Using a pre-existing model is off the table if it incurs a patent claim.

1. Must provide access to the 3rd party websites to audit and monitor the authenticity and behaviour of the service.
2. Enabling open-source, academic and research communities to audit the Algorithms and research the efficacy of the product.
3. Laws controlling data collection: Some websites might have a policy against collecting customer data in form of reviews and ratings.
4. Must be responsible for the scraped data. It is quintessential to protect the privacy and intention with which the data was extracted.

8. Applicable Constraints.

1. The use of cloud platforms to store the data gathered over the net.
2. Using the spark service to clean and transform data.
3. For Evaluation of the model which is done with the help of tableau and PowerBI.
4. For modelling Logistic Regression and Decision Tree classifiers are applied.

9. Business Opportunity.

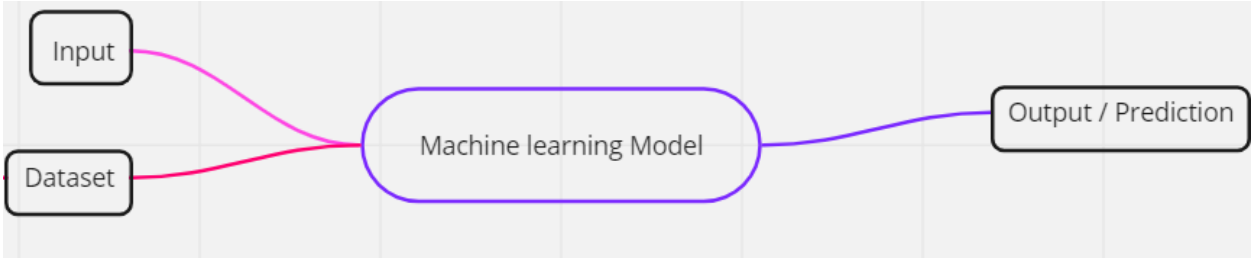
9.1. Patients with positive results can be referred to doctors and hospitals that have a tie-up with our app. Any medical personnel or organization wanting to register will have to pay a certain percentage of the visiting/checkup fees to the app as we are sending them those patients.

9.2. An organizational level of the application can be developed which will be paid to use and can be used only by medical organizations like hospitals, medical research institutes, etc.

10. Concept Generation.

This product requires the tool of machine learning models to be written from scratch in order to suit our needs. . Tweaking these models for our use is less daunting than coding them up from scratch. A well-trained model can either be repurposed or built. But building a model with the resources and data we have is dilatory but possible. The customer might want to spend the least amount of time giving input data. . This accuracy will take a little effort to nail because it's imprudent to rely purely on the Classic Machine Learning algorithm.

In this prototype, I will use Logistic Regression and Decision Tree classifier and compare both results.



Changing object data to integer values.

```
heart_data =  heart_data[heart_data.columns].replace({'Yes':1, 'No':0, 'Male':1,'Female':0,'No, borderline diabetes':'0','Yes (during pregnancy)':'1' })
heart_data['Diabetic'] = heart_data['Diabetic'].astype(int)

```

✓ 1.9s

Splitting the data into train and test sets

```
features=heart_data.drop(columns=['HeartDisease'])
target=heart_data['HeartDisease']

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(features, target, shuffle = True, test_size = .2, random_state = 44)

```

✓ 0.2s Python

One-Hot Encoding

```
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import make_column_transformer
transformer = make_column_transformer(
    (OneHotEncoder(sparse=False), ['AgeCategory', 'Race', 'GenHealth']),
    remainder='passthrough')

# Encode training data
transformed_train = transformer.fit_transform(X_train)
transformed_train_data = pd.DataFrame(transformed_train, columns=transformer.get_feature_names())

# Concat the two tables
transformed_train_data.reset_index(drop=True, inplace=True)
X_train.reset_index(drop=True, inplace=True)
X_train = pd.concat([transformed_train_data, X_train], axis=1)

# Remove old columns
X_train.drop(['AgeCategory', 'Race', 'GenHealth'], axis = 1, inplace = True)



# Encode test data
transformed_test = transformer.fit_transform(X_test)
transformed_test_data = pd.DataFrame(transformed_test, columns=transformer.get_feature_names())
```

```
# Concat the two tables
transformed_test_data.reset_index(drop=True, inplace=True)
X_test.reset_index(drop=True, inplace=True)
X_test = pd.concat([transformed_test_data, X_test], axis=1)


# Remove old columns
X_test.drop(['AgeCategory', 'Race', 'GenHealth'], axis = 1, inplace = True)
```

✓ 0.7s

Standard Scaling

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

# Scale trainint data
X_train = scaler.fit_transform(X_train)

# Scale test data
X_test = scaler.fit_transform(X_test)
```

✓ 0.5s

Decision tree Classifier.

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix
model=DecisionTreeClassifier(criterion='entropy', max_depth=3).fit(X_train,y_train)
pred=model.predict(X_test)
# pred_arr=[y[i] for i in pred]
# print(pred_arr)
acc=model.score(X_test,y_test)
confusion_matrix_dtrc = confusion_matrix(y_test,pred)
print("Accuracy of the model=",acc*100,"%")
print("The confusion matrix: \n",confusion_matrix_dtrc)
```

✓ 0.8s

```
Accuracy of the model= 91.52113072437031 %
The confusion matrix:
 [[58427    86]
 [ 5337   109]]
```

Logistic regression model.

```
1  from sklearn.linear_model import LogisticRegression
2  from sklearn.metrics import confusion_matrix
3  lr=LogisticRegression(penalty='l2',solver="newton-cg")
4  lr.fit(X_train,y_train)
5  pred=lr.predict(X_test)
6  acc=lr.score(X_test,y_test)
7  conf=confusion_matrix(y_test,pred)
8
9  print("Accuracy:",acc*100)
10 print("Confusion MAtrix:\n",conf)
11
```

✓ 4.8s

```
Accuracy: 91.6118138182273
Confusion MAtrix:
 [[58003   510]
 [ 4855   591]]
```

11. Final Report Prototype.

The product makes use of these operations:

Back end:

Model Development: This must be done before releasing the service. A lot of manual supervised machine learning must be performed to optimize automated tasks.

1. Performing EDA to realize the dependent and independent features.

2. Algorithm training and optimization must be done to minimize overfitting of the model and hyperparameter tuning.

Front end:

1. Different user interface: The user must be given many options to choose from in terms of parameters. This can only be optimized after a lot of testing and analysis of all the edge

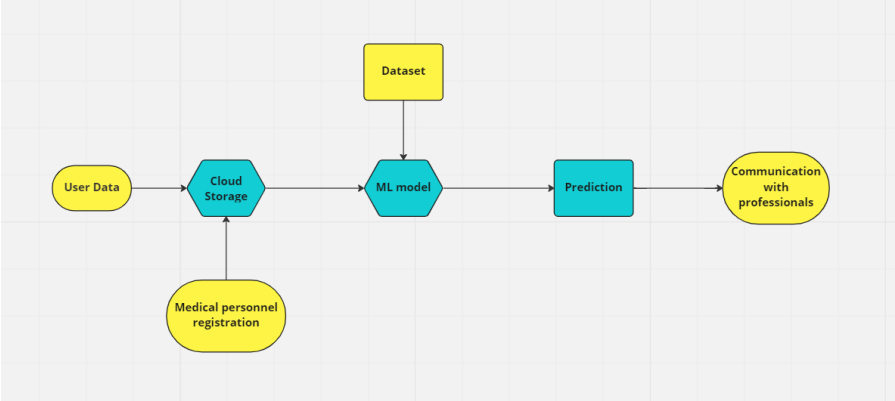
cases.

2. Interactive visualization of the data extracted from the trained models will return raw and inscrutable data. This must be present in an aesthetic and an “easy to read” style.

3. Feedback system: A valuable feedback system must be developed to understand the customer’s needs that have not been met. This will help us train the models constantly.

12. Product details: Working.

The diagram below shows a brief prototype of the final model. The application takes registration in two modes: 1. User 2. Medical personnel. The data is then stored in the cloud storage which can be accessed by the Machine Learning model in the back end. The model is trained using some datasets from which the model makes predictions. After this, if the user wants to consult a professional they can do so through the app.



13. Conclusion.

Artificial Intelligence has touched almost every aspect of human life and medicine is one of the most important parts. AI is set to change the medical industry in the coming decades — it wouldn’t make sense for pathology to not be disrupted too. Currently, ML models are still in the testing and experimentation phase for medical purposes. As datasets are getting larger and of

higher quality, researchers are building highly accurate models. Hence a working model is bound to gain the attention of some keen eyes.

14. References.

1. Heart Disease facts (cdc.gov): https://www.cdc.gov/heartdisease/facts.htm

2. Cardio Vascular Diseases (WHO): https://www.who.int/health-topics/cardiovascular-diseases

3. Heart Disease Dataset (Kaggle): https://www.kaggle.com/code/andls555/heart-disease-prediction/data

4. Patent information: https://www.wipo.int/ipadvantage/en/details.jsp?id=12463